Self-Attentive, Multi-Context One-Class Classification for Unsupervised Anomaly Detection on Text

Lukas Ruff† Yury Zemlyanskiy‡
Robert Vandermeulen§ Thomas Schnake† Marius Kloft§‡
†Technical University of Berlin, Berlin, Germany
‡University of Southern California, Los Angeles, United States
§Technical University of Kaiserslautern, Kaiserslautern, Germany
{lukas.ruff,t.schnake}@tu-berlin.de yury.zemlyanskiy@usc.edu
{vandermeulen,kloft}@cs.uni-kl.de

Abstract

There exist few text-specific methods for unsupervised anomaly detection, and for those that do exist, none utilize pre-trained models for distributed vector representations of words. In this paper we introduce a new anomaly detection method—Context Vector Data Description (CVDD)—which builds upon word embedding models to learn multiple sentence representations that capture multiple semantic contexts via the self-attention mechanism. Modeling multiple contexts enables us to perform contextual anomaly detection of sentences and phrases with respect to the multiple themes and concepts present in an unlabeled text corpus. These contexts in combination with the self-attention weights make our method highly interpretable. We demonstrate the effectiveness of CVDD quantitatively as well as qualitatively on the well-known Reuters, 20 Newsgroups, and IMDB Movie Reviews datasets.

1 Introduction

Anomaly Detection (AD) (Chandola et al., 2009; Pimentel et al., 2014; Aggarwal, 2017) is the task of discerning rare or unusual samples in a corpus of unlabeled data. A common approach to AD is one-class classification (Moya et al., 1993), where the objective is to learn a model that compactly describes “normality”—usually assuming that most of the unlabeled training data is “normal,” i.e. non-anomalous. Deviations from this description are then deemed to be anomalous. Examples of one-class classification methods are the well-known One-Class SVM (OC-SVM) (Schölkopf et al., 2001) and Support Vector Data Description (SVDD) (Tax and Duin, 2004).

Applying AD to text is useful for many applications including discerning anomalous web content (e.g. posts, reviews, or product descriptions), automated content management, spam detection, and characterizing news articles so as to identify similar or dissimilar novel topics. Recent work has found that proper text representation is critical for designing well-performing machine learning algorithms. Given the exceptional impact that universal vector embeddings of words (Bengio et al., 2003) such as word2vec (Mikolov et al., 2013), GloVe (Pennington et al., 2014), and FastText (Bojanowski et al., 2017; Joulin et al., 2017) or dynamic vector embeddings of text by language models such as ELMo (Peters et al., 2018) and BERT (Devlin et al., 2018) have had on NLP, it is somewhat surprising that there has been no work on adapting AD techniques to use such unsupervised pre-trained models. Existing AD methods for text still typically rely on bag-of-words (BoW) text representations (Manevitz and Yousef, 2001, 2007; Mahapatra et al., 2012; Kannan et al., 2017).

In this work, we introduce a novel one-class classification method that takes advantage of pre-trained word embedding models for performing AD on text. Starting with pre-trained word embeddings, our method—Context Vector Data Description (CVDD)—finds a collection of transforms to map variable-length sequences of word embeddings to a collection of fixed-length text representations via a multi-head self-attention mechanism. These representations are trained along with a collection of context vectors such that the context vectors and representations are similar while keeping the context vectors diverse. Training these representations and context vectors jointly allows our algorithm to capture multiple modes of normalcy which may, for example, correspond to a collection of distinct yet non-anomalous topics. Disentangling such modes allows for contextual anomaly detection with sample-based explanations and enhanced model interpretability.
2 Context Vector Data Description

In this section we introduce Context Vector Data Description (CVDD), a self-attentive, multi-context one-class classification method for unsupervised AD on text. We first describe the CVDD model and objective, followed by a description of its optimization procedure. Finally we present some analysis of CVDD.

2.1 Multi-Head Self-Attention

Here we describe the problem setting and multi-head self-attention mechanism which lies at the core of our method. Let \( S = (w_1, \ldots, w_\ell) \in \mathbb{R}^{d \times \ell} \) be a sentence or, more generally, a sequence of \( \ell \in \mathbb{N} \) words (e.g. phrase or document), where each word is represented via some \( d \)-dimensional vector. Given some pre-trained word embedding, let \( H = (h_1, \ldots, h_\ell) \in \mathbb{R}^{p \times \ell} \) be the corresponding \( p \)-dimensional vector embeddings of the words in \( S \). The vector embedding \( H \) might be some universal word embedding (e.g. GloVe, FastText) or the hidden vector activations of sentence \( S \) given by some language model (e.g. ELMo, BERT).

The aim of multi-head self-attention (Lin et al., 2017) is to define a transformation that accepts sentences \( S^{(1)}, \ldots, S^{(n)} \) of varying lengths \( \ell^{(1)}, \ldots, \ell^{(n)} \) and returns a vector of fixed length, thereby allowing us to apply more standard AD techniques. The idea here is to find a fixed-length vector representation of size \( p \) via a convex combination of the word vectors in \( H \). The coefficients of this convex combination are adaptive weights which are learned during training.

We describe the model now in more detail. Given the word embeddings \( H \in \mathbb{R}^{p \times \ell} \) of a sentence \( S \), the first step of the self-attention mechanism is to compute the attention matrix \( A \in (0, 1)^{r \times \ell} \) by

\[
A = \text{softmax} \left( \tanh(H^T W_1) W_2 \right),
\]

where \( W_1 \in \mathbb{R}^{p \times d_a} \) and \( W_2 \in \mathbb{R}^{d_a \times r} \). The \( \tanh \)-activation is applied element-wise and the \text{softmax} column-wise, thus making each vector \( a_k \) of the attention matrix \( A = (a_1, \ldots, a_r) \) a positive vector that sums to one, i.e. a weighting vector. The \( r \) vectors \( a_1, \ldots, a_r \) are called attention heads with each head giving a weighting over the words in the sentence. Dimension
$d_a$ is the internal dimensionality and thus sets the complexity of the self-attention module. We now obtain a fixed-length sentence embedding matrix $M = (m_1, \ldots, m_r) \in \mathbb{R}^{p \times r}$ from the word embeddings $H$ by applying the self-attention weights $A$ as

$$M = HA.$$  

Thus, each column $m_k \in \mathbb{R}^p$ is a weighted linear combination of the vector embeddings $h_1, \ldots, h_r \in \mathbb{R}^p$ with weights $a_k \in \mathbb{R}^l$ given by the respective attention head $k$, i.e. $m_k = Ha_k$. Often, a regularization term $P$ such as

$$P = \frac{1}{n} \sum_{i=1}^{n} \| (A^{(i)})^T A^{(i)} - I \|_F^2$$  

is added to the learning objective to promote the attention heads to be nearly orthogonal and thus capture distinct views that focus on different semantics and concepts of the data. Here, $I$ denotes the $r \times r$ identity matrix, $\| \cdot \|_F$ is the Frobenius norm, and $A^{(i)} \triangleq A(H^{(i)}; W_1, W_2)$ is the attention matrix corresponding to sample $S^{(i)}$.

### 2.2 The CVDD Objective

In this section, we introduce an unsupervised AD method for text. It aims to capture multiple distinct contexts that may appear in normal text. To do so, it leverages the multi-head self-attention mechanism (described in the previous section), with the heads focusing on distinct contexts (one head per context).

We first define a notion of similarity. Let $\text{sim}(u, v)$ be the cosine similarity between two vectors $u$ and $v$, i.e.

$$\text{sim}(u, v) = \frac{\langle u, v \rangle}{\|u\|\|v\|} \in [-1, 1]$$  

and denote by $d(u, v)$ the cosine distance between $u$ and $v$, i.e.

$$d(u, v) = \frac{1}{2} (1 - \text{sim}(u, v)) \in [0, 1].$$  

As before, let $r$ be the number of attention heads. We now define the context matrix $C = (c_1, \ldots, c_r) \in \mathbb{R}^{p \times r}$ to be a matrix whose columns $c_1, \ldots, c_r$ are vectors in the word embedding space $\mathbb{R}^p$. Given an unlabeled training corpus $S^{(1)}, \ldots, S^{(n)}$ of sentences (or phrases, documents, etc.), which may vary in length $\ell^{(i)}$, and their corresponding word vector embeddings $H^{(1)}, \ldots, H^{(n)}$, we formulate the Context Vector Data Description (CVDD) objective as follows:

$$\min_{C, W_1, W_2} \frac{1}{n} \sum_{i=1}^{n} \sum_{k=1}^{r} \sigma_k(H^{(i)}) d(c_k, m_k^{(i)}).$$  

$\sigma_1(H), \ldots, \sigma_r(H)$ are input-dependent weights, i.e. $\sum_k \sigma_k(H) = 1$, which we specify further below in detail. That is, CVDD finds a set of vectors $c_1, \ldots, c_r \in \mathbb{R}^p$ with small cosine distance to the attention-weighted data representations $m_1^{(i)}, \ldots, m_r^{(i)} \in \mathbb{R}^p$. Note that each concept vector $c_k$ is paired with the data representation $m_k^{(i)}$ of the $k$th head. This causes the network to learn attention weights that capture diverse concepts and themes from the data. We call the vectors $c_1, \ldots, c_r \in \mathbb{R}^p$ context vectors because they represent a compact description of the different contexts that are present in the data. For a given text sample $S^{(i)}$, the corresponding embedding $m_k^{(i)}$ provides a sample representation with respect to the $k$th context.

**Multi-context regularization** To promote the context vectors $c_1, \ldots, c_r$ to capture diverse themes and concepts, we regularize them towards orthogonality:

$$P(C) = \|C^T C - I\|_F^2.$$  

We can now state the overall CVDD objective as

$$\min_{C, W_1, W_2} J_n(C, W_1, W_2) + \lambda P(C),$$  

where $J_n(C, W_1, W_2)$ is the objective function from Eq. (6) and $\lambda > 0$. Because CVDD minimizes the cosine distance

$$d(c_k, m_k) = \frac{1}{2} \left( 1 - \frac{\langle c_k, m_k \rangle}{\|c_k\|\|m_k\|} \right),$$  

regularizing the context vectors $c_1, \ldots, c_k$ to be orthogonal implicitly regularizes the attention weight vectors $a_1, \ldots, a_r$ to be orthogonal as well, a phenomenon which we also observed empirically. Despite this, we found that regularizing the context vectors as in (7) allows for faster, more stable optimization in comparison to regularizing the attention weights as in (3). We suspect this is because in (3) $P = P_n(W_1, W_2)$ depends nonlinearly on the attention network weights $W_1$ and $W_2$ as well as on the data batches. In comparison, the gradients of $P(C)$ in (7) can be directly
computed. Empirically we found that selecting \( \lambda \in \{1, 10\} \) gives reliable results with the desired effect that CVDD learns multiple distinct contexts.

**Optimization of CVDD** We optimize the CVDD objective jointly over the self-attention network weights \( \{W_1, W_2\} \) and the context vectors \( c_1, \ldots, c_r \) using stochastic gradient descent (SGD) and its variants (e.g. Adam (Kingma and Ba, 2014)). Thus, CVDD training scales linearly in the number of training batches. Training is carried out until convergence. Since the self-attention module is just a two-layer feed-forward network, the computational complexity of training CVDD is very low. However, evaluating a pre-trained model for obtaining word embeddings may add to the computational cost (e.g. in case of large pre-trained language models) in which case parallelization strategies (e.g. by using GPUs) should be exploited. We initialize the context vectors with the centroids from running \( k\)-means++ (Arthur and Vassilvitskii, 2007) on the sentence representations obtained from averaging the word embeddings. We empirically found that this initialization strategy improves optimization speed as well as performance.

**Weighting contexts in the CVDD objective** There is a natural motivation to consider multiple vectors for representation because sentences or documents may have multiple contexts, e.g. cinematic language, movie theme, or sentiment in movie reviews. This raises the question of how these context representations should be weighted in a learning objective. For this, we propose to use a parameterized softmax over the \( r \) cosine distances of a sample \( S \) with embedding \( H \) in our CVDD objective:

\[
\sigma_k(H) = \frac{\exp(-\alpha \, d(c_k, m_k(H)))}{\sum_{j=1}^r \exp(-\alpha \, d(c_j, m_j(H)))},
\]

for \( k = 1, \ldots, r \) with \( \alpha \in [0, +\infty) \). The \( \alpha \) parameter allows us to balance the weighting between two extreme cases: (i) \( \alpha = 0 \) which results in all contexts being equally weighted, i.e. \( \sigma_k(H) = 1/r \) for all \( k \), and (ii) \( \alpha \to \infty \) in which case the softmax approximates the argmin-function, i.e. only the closest context \( k_{\text{min}} = \arg\min_k d(c_k, m_k) \) has weight \( \sigma_{k_{\text{min}}} = 1 \) whereas \( \sigma_k = 0 \) for \( k \neq k_{\text{min}} \) otherwise.

Traditional clustering methods typically only consider the argmin, i.e. the closest representatives (e.g. nearest centroid in \( k\)-means). For learning multiple sentence representations as well as contexts from data, however, this might be ineffective and result in poor representations. This is due to the optimization getting “trapped” by the closest context vectors which strongly depend on the initialization. Not penalizing the distances to other context vectors also does not foster the extraction of multiple contexts per sample. For this reason, we initially set \( \alpha = 0 \) in training and then gradually increase the \( \alpha \) parameter using some annealing strategy. Thus, learning initially focuses on extracting multiple contexts from the data before sample representations then gradually get fine-tuned w.r.t their closest contexts.

2.3 Contextual Anomaly Detection

Our CVDD learning objective and the introduction of context vectors allows us to score the “anomalousness” in relation to these various contexts, i.e. to determine anomalies contextually. We naturally define the anomaly score w.r.t. context \( k \) for some sample \( S \) with embedding \( H \) as

\[
s_k(H) = d(c_k, m_k(H)),
\]

the cosine distance of the contextual embedding \( m_k(H) \) to the respective context vector \( c_k \). A greater the distance of \( m_k(H) \) to \( c_k \) implies a more anomalous sample w.r.t. context \( k \). A straightforward choice for an overall anomaly score then is to take the average over the contextual anomaly scores:

\[
s(H) = \frac{1}{r} \sum_{k=1}^r s_k(H).
\]

One might, however, select different weights for different contexts as particular contexts might be more or less relevant in certain scenarios. Using word lists created from the most similar attention-weighted sentences to a context vector provides an interpretation of the particular context.

2.4 Avoiding Manifold Collapse

Neural approaches to AD (Ruff et al., 2018) and clustering (Fard et al., 2018) are prone to converge to degenerate solutions where the data is transformed to a small manifold or a single point. CVDD potentially may also suffer from this manifold collapse phenomenon. Indeed, there exists a theoretical optimal solution (\( C^*, W_1^*, W_2^* \)) for which the (nonnegative) CVDD objective (6) becomes zero due to trivial representations. This is
the case for \((C^*, W_1^*, W_2^*)\) where

\[
m_k(H^{(i)}; W_1^*, W_2^*) = c_k^i \quad \forall i = 1, \ldots, n, (13)
\]

holds, i.e. if the contextual representation \(m_k(\ ; W_1^*, W_2^*)\) is a constant mapping. In this case, all contextual data representations have collapsed to the respective context vectors and are independent of the input sentence \(S\) with embedding \(H\). Because the pre-trained embeddings \(H\) are fixed, and the self-attention embedding must be a convex combination of the columns in \(H\), it is difficult for the network to overfit to a constant function. A degenerate solution may only occur if there exists a word which occurs in the same location in all training samples. This property of CVDD, however, might be used “as a feature” to uncover such simple common structure amongst the data such that appropriate pre-processing steps can be carried out to rule out such “Clever Hans” behavior (Lapuschkin et al., 2019). Finally, since we normalize the contextual representations \(m_k\) as well as the context vectors \(c_k\) with cosine similarity, a trivial collapse to the origin \((m_k = 0\) or \(c_k = 0\)) is also avoided.

3 Related Work

Our method is related to works from unsupervised representation learning for NLP, methods for AD on text, as well as representation learning for AD.

Vector representations of words (Bengio et al., 2003; Collobert and Weston, 2008) or word embeddings have been the key for many substantial advances in NLP in recent history. Well-known examples include word2vec (Mikolov et al., 2013), GloVe (Pennington et al., 2014), or fastText (Bojanowski et al., 2017; Joulin et al., 2017). Approaches for learning sentence embeddings have also been introduced, including SkipThought (Kiros et al., 2015), ParagraphVector (Le and Mikolov, 2014), Conceptual Sentence Embedding (Wang et al., 2016), Sequential Denoising Autoencoders (Hill et al., 2016) or FastSent (Hill et al., 2016). In a comparison of unsupervised sentence embedding models, Hill et al. (2016) show that the optimal embedding critically depends on the targeted downstream task. For specific applications, more complex deep models such as recurrent (Chung et al., 2014), recursive (Socher et al., 2013) or convolutional (Kim, 2014) networks that learn task-specific dynamic sentence embeddings usually perform best. Recently, large language models like ELMo (Peters et al., 2018) or BERT (Devlin et al., 2018) that learn dynamic sentence embeddings in an unsupervised manner have proven to be very effective for transfer learning, beating the state-of-the-art in many downstream tasks. Such large deep models, however, are very computationally intensive. Finally, no method for learning representations of words or sentences specifically for the AD task have been proposed yet.

There are only few works addressing AD on text. Manevitz and Yousef study one-class classification of documents using the OC-SVM (Schölkopf et al., 2001; Manevitz and Yousef, 2001) and a simple autoencoder (Manevitz and Yousef, 2007). Liu et al. (2002) consider learning from positively labeled as well as unlabeled mixed data of documents what they call a “partially supervised classification” problem that is similar to one-class classification. Kannan et al. (2017) introduce a non-negative matrix factorization method for AD on text that is based on block coordinate descent optimization. Mahapatra et al. (2012) include external contextual information for detecting anomalies using LDA clustering. All the above works, however, only consider document-to-word co-occurrence text representations. Other approaches rely on specific handcrafted features for particular domains or types of anomalies (Guthrie, 2008; Kumaraswamy et al., 2015). None of the existing methods make use of pre-trained word models that were trained on huge corpora of text.

Learning representations for AD or Deep Anomaly Detection (Chalapathy and Chawla, 2019) has seen great interest recently. Such approaches are motivated by applications on large and complex datasets and the limited scalability of classic, shallow AD techniques and their need for manual feature engineering. Deep approaches aim to overcome those limitations by automatically learning relevant features from the data and mini-batch training for improved computational scaling. Most existing deep AD works are in the computer vision domain and show promising, state-of-the-art results for image data (Andrews et al., 2016; Schlegl et al., 2017; Ruff et al., 2018; Deecke et al., 2018; Golan and El-Yaniv, 2018; Hendrycks et al., 2019). Other works examine deep AD on general high-dimensional point data.
(Sakurada and Yairi, 2014; Xu et al., 2015; Erfani et al., 2016; Chen et al., 2017). Few deep approaches examine sequential data, and those that do exist focus on time series data AD using LSTM networks (Bontemps et al., 2016; Malhotra et al., 2015, 2016). As mentioned earlier, there exists no representation learning approach for AD on text.

4 Experiments

We evaluate the performance of CVDD quantitatively in one-class classification experiments on the Reuters-21578\footnote{daviddewis.com/resources/testcollections/reuters21578} and 20 Newsgroups\footnote{gwone.com/jason/20Newsgroups} datasets as well as qualitatively in an application on IMDB Movie Reviews\footnote{ai.stanford.edu/amasa/data/sentiment} to detect anomalous reviews.\footnote{Code available at: github.com/lukasruh/CVDD-PyTorch}

4.1 Experimental Details

Pre-trained Models

We employ the pre-trained GloVe (Pennington et al., 2014) as well as fastText (Bojanowski et al., 2017; Joulin et al., 2017) word embeddings in our experiments. For GloVe we consider the 6B token vector embeddings of \( p = 300 \) dimensions which have been trained on the Wikipedia and Gigaword 5 corpora. For fastText we consider the English word vectors that also have \( p = 300 \) dimensions which have been trained on the Wikipedia and English webcrawl. We also experimented with dynamic word embeddings from the BERT (Devlin et al., 2018) language model but did not observe improvements over GloVe or fastText on the considered datasets that would justify the added computational cost.

Baselines

We consider three baselines for aggregating word vector embeddings to fixed-length sentence representations: (i) mean, (ii) tf-idf weighted mean, and (iii) max-pooling. It has been repeatedly observed that the simple average sentence embedding proves to be a strong baseline in many tasks (Wieting et al., 2016; Arora et al., 2017; Rücklé et al., 2018). Max-pooling is commonly applied over hidden activations (Lee and Dernoncourt, 2016). The tf-idf weighted mean is a natural sentence embedding baseline that includes document-to-term co-occurrence statistics. For AD, we then consider the OC-SVM (Schölkopf et al., 2001) with cosine kernel (which in this case is equivalent to SVDD (Tax and Duin, 2004)) on these sentence embeddings where we always train for hyperparameters \( \nu \in \{0.05, 0.1, 0.2, 0.5\} \) and report the best result.

CVDD configuration

We employ self-attention with \( d_a = 150 \) for CVDD and present results for \( r \in \{3, 5, 10\} \) number of attention heads. We use Adam (Kingma and Ba, 2014) with a batch size of 64 for optimization and first train for 40 epochs with a learning rate of \( \eta = 0.01 \) after which we train another 60 epochs with \( \eta = 0.001 \), i.e. we establish a simple two-phase learning rate schedule. For weighting contexts, we consider the case of equal weights \( (\alpha = 0) \) as well as a logarithmic annealing strategy \( \alpha \in \{0, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}\} \) where we update \( \alpha \) every 20 epochs. For multi-context regularization, we choose \( \lambda \in \{1, 10\} \).

Data pre-processing

On all three datasets, we always lowercase text and strip punctuation, numbers, as well as redundant whitespaces. Moreover, we remove stopwords using the stopwords list from the nltk library (Bird et al., 2009) and only consider words with a minimum length of 3 characters.

4.2 One-Class Classification of News Articles

Setup

We perform one-class classification experiments on the Reuters-21578 and 20 Newsgroups topic classification datasets which allow us to quantitatively evaluate detection performance via AUC measure by using the ground truth labels in testing. Such use of classification datasets is the typical evaluation approach in the AD literature (Erfani et al., 2016; Ruff et al., 2018; Golan and El-Yaniv, 2018). For the multi-label Reuters dataset, we only consider the subset of samples that have exactly one label and have selected the classes such that there are at least 100 training examples per class. For 20 Newsgroups, we consider the six top-level subject matter groups computer, recreation, science, miscellaneous, politics, and religion as distinct classes. In every one-class classification setup, one of the classes is the normal class and the remaining classes are considered anomalous. We always train the models only with the training data from the respective normal class. We then perform testing on the test samples from all classes, where samples from the normal class get label \( y = 0 \) (“normal”) and samples from all the remaining classes get label \( y = 1 \) (“anomalous”) for determining the AUC.
| Normal Class | GloVe | fastText | OC-SVM | CVDD |
|-------------|-------|----------|--------|------|
|             | mean  | tf-idf  | max    | r=3  | r=5  | r=10 | c*     | mean  | tf-idf  | max    | r=3  | r=5  | r=10 | c*     |
| Reuters     |       |         |        |      |      |      |        |       |         |        |      |      |      |        |
| earn        | 91.1  | 88.6    | 77.1   | 94.0 | 92.8 | 91.8 | 97.6   | 91.6  | 82.4    | 74.9  | 95.3 | 92.7 | 93.9 | 94.5   |
| acq         | 93.1  | 77.0    | 81.4   | 90.2 | 88.7 | 91.5 | 95.6   | 91.8  | 74.1    | 80.2  | 91.0 | 90.3 | 92.7 | 92.4   |
| crude       | 92.0  | 90.3    | 91.2   | 89.6 | 92.5 | 95.5 | 89.4   | 93.3  | 90.2    | 84.7  | 90.9 | 94.1 | 97.3 | 85.0   |
| trade       | 99.0  | 96.8    | 93.7   | 98.3 | 98.2 | 99.2 | 97.9   | 97.6  | 95.0    | 92.1  | 97.9 | 98.1 | 99.3 | 97.7   |
| money-fx    | 88.6  | 81.2    | 73.6   | 82.5 | 76.7 | 82.8 | 99.7   | 80.5  | 82.6    | 73.8  | 82.6 | 79.8 | 82.5 | 99.5   |
| interest    | 97.4  | 93.5    | 84.2   | 92.3 | 91.7 | 97.7 | 98.4   | 91.6  | 88.7    | 82.8  | 93.3 | 92.1 | 95.9 | 97.4   |
| ship        | 91.2  | 95.1    | 85.6   | 97.6 | 96.9 | 95.6 | 99.7   | 90.0  | 90.6    | 85.0  | 96.9 | 94.7 | 96.1 | 99.7   |

Table 1: AUCs (in %) of one-class classification experiments on Reuters and 20 Newsgroups.

### Example of top words per context on 20 Newsgroups

| computer | politics | religion |
|----------|----------|----------|
| get      | kill     | example  |
| help     | killed   | particular |
| thanks   | escape   | certain  |
| appreciated | away    | political |
| got      | back     | analysis  |
| know     | shoot    | forces   |
| way      | shot     | support  |
| try      | crying   | therefore |
| tried    | killing  | movement |
| take     | understand | leaders  |
|          |         |         |

Table 2: Example of top words per context on 20 Newsgroups one-class classification experiments comp, pol, and rel for CVDD model with $r = 3$ contexts.

### Top words of the best single contexts $c^*$ for one-class classification experiments of news articles

Table 3: Top words of the best single contexts $c^*$ for one-class classification experiments of news articles.

**Results** The results are presented in Table 1. Overall, we can see that CVDD shows competitive detection performance. We compute the AUCs for CVDD from the average anomaly score over the contextual anomaly scores as defined in (12). We find CVDD performance to be robust over $\lambda \in \{1, 10\}$ and results are similar for weighting contexts equally ($\alpha = 0$) or employing the logarithmic annealing strategy. The CVDD results in Table 1 are averages over those hyperparameters.

We get an intuition of the theme captured by some CVDD context vector by examining a list of top words for this context. We create such lists by counting the top words according to the highest self-attention weights from the most similar test sentences per context vector. Table 2 shows an example of such context word lists for CVDD three contexts. Such lists may guide a user in weighting and selecting relevant contexts in a specific application. Following this thought, we also report the
4.3 Detecting Anomalous Movie Reviews

Setup We apply CVDD for detecting anomalous reviews in a qualitative experiment on IMDB Movie Reviews. For this, we train a CVDD model with \( r = 10 \) context vectors on the full IMDB train set of 25,000 movie review samples. After training, we examine the most anomalous and most normal reviews according to the CVDD anomaly scores on the IMDB test set which also includes 25,000 reviews. We use GloVe word embeddings and keep the CVDD model settings and training details as outlined in Section 4.1.

Results Table 4 shows the top words for each of the \( r = 10 \) CVDD contexts of the trained model. We can see that the different contexts of the CVDD model capture the different themes present in the movie reviews well. Note for example that \( c_1 \) and \( c_2 \) represent positive and negative sentiments respectively, \( c_3, c_7, \) and \( c_{10} \) refer to different aspects of cinematic language, and \( c_9 \) captures names, for example. Figure 1 in the introduction depicts qualitative examples of this experiment. 1a are the movie reviews having the highest anomaly scores. The top anomaly is a review that repeats the same phrase many times. From examining the most anomalous reviews, the dataset seems to be quite clean in general though. Figure 1c shows the most normal reviews w.r.t. the first three contexts, i.e. the samples that have the lowest respective contextual anomaly scores. Here, the self-attention weights give a sample-based explanation for why a particular review is normal in view of the respective context.

Table 4: Top words per context on IMDB Movie Reviews for CVDD model with \( r = 10 \) contexts.

| \( c_1 \) | \( c_2 \) | \( c_3 \) | \( c_4 \) | \( c_5 \) | \( c_6 \) | \( c_7 \) | \( c_8 \) | \( c_9 \) | \( c_{10} \) |
|---|---|---|---|---|---|---|---|---|---|
| great | awful | plot | two | think | actions | film | head | william | movie |
| excellent | downright | characters | one | anybody | development | filmmakers | back | john | movies |
| good | stupid | story | three | know | efforts | filmmaker | onto | michael | porn |
| superb | inept | storyline | first | would | establishing | movie | cut | richard | sex |
| well | pathetic | scenes | five | say | knowledge | syberberg | bottom | davis | watch |
| wonderful | irritating | narrative | four | really | involvement | cinema | neck | david | teen |
| nice | annoying | subplots | five | part | want | policies | director | floor | james | best |
| best | inane | twists | every | never | individuals | acting | flat | walker | dvd |
| terrific | unfunny | tale | best | suppose | necessary | filmmaking | thick | robert | scenes |
| beautiful | horrible | interesting | also | actually | concerning | actors | front | gordon | flick |

We presented a new self-attentive, multi-context one-class classification approach for unsupervised anomaly detection on text which makes use of pre-trained word models. Our method, Context Vector Data Description (CVDD), enables contextual anomaly detection and has strong interpretability features. We demonstrated the detection performance of CVDD empirically and showed qualitatively that CVDD is well capable of learning distinct, diverse contexts from unlabeled text corpora.

Acknowledgments

We thank Fei Sha for insightful discussions and comments as well as the anonymous reviewers for their helpful suggestions. LR acknowledges support from the German Federal Ministry of Education and Research (BMBF) in the project ALICE III (FKZ: 01IS18049B). YZ is partially supported by NSF Awards IIS-1513966, IIS-1632803, IIS-1833137, CCF-1139148, DARPA Award FA8750-18-2-0117, DARPA-D3M Award UCB-00009528, Google Research Awards, gifts from Facebook...
and Netflix, and ARO W911NF-12-1-0241 and W911NF-15-1-0484. MK and RV acknowledge support by the German Research Foundation (DFG) award KL 2698/2-1 and by the German Federal Ministry of Education and Research (BMBF) awards 031L0023A, 01IS18051A, and 031B0770E. Part of the work was done while MK was a sabbatical visitor of the DASH Center at the University of Southern California.

References

Charu C. Aggarwal. 2017. *Outlier Analysis*. Springer.

J. T. A. Andrews, E. J. Morton, and L. D. Griffin. 2016. Detecting Anomalous Data Using Auto-Encoders. *IJMLC*, 6(1):21.

Sanjeev Arora, Yingyu Liang, and Tengyu Ma. 2017. A simple but tough-to-beat baseline for sentence embeddings. In *ICLR*.

David Arthur and Sergei Vassilvitskii. 2007. k-means++. The advantages of careful seeding. In *ACM-SIAM symposium on discrete algorithms*, pages 1027–1035.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. 2003. A neural probabilistic language model. *JMLR*, 3(Feb):1137–1155.

Steven Bird, Edward Loper, and Ewan Klein. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit*. O’Reilly Media, Inc.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Loïc Bontemps, James McDermott, Nhien-An Le-Khac, et al. 2016. Collective anomaly detection based on long short-term memory recurrent neural networks. In *International Conference on Future Data and Security Engineering*, pages 141–152. Springer.

Raghavendra Chalapathy and Sanjay Chawla. 2019. Deep learning for anomaly detection: A survey. *arXiv:1901.03407*.

V. Chandola, A. Banerjee, and V. Kumar. 2009. Anomaly Detection: A Survey. *ACM Computing Surveys*, 41(3):1–58.

J. Chen, S. Sathe, C. Aggarwal, and D. Turaga. 2017. Outlier Detection with Autoencoder Ensembles. In *SDM*, pages 90–98.

Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. In *NIPS Workshop*.

Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *ICML*, pages 160–167.

Lucas Deecke, Robert A. Vandermeulen, Lukas Ruff, Stephan Mandt, and Marius Kloft. 2018. Image anomaly detection with generative adversarial networks. In *ECML-PKDD*.

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

S. M. Erfani, S. Rajasegarar, S. Karunasekera, and C. Leckie. 2016. High-dimensional and large-scale anomaly detection using a linear one-class SVM with deep learning. *Pattern Recognition*, 58:121–134.

Maziar Moradi Fard, Thibaut Thonet, and Eric Gaussier. 2018. Deep k-means: Jointly clustering with k-means and learning representations. *arXiv:1806.10069*.

Izhak Golan and Ran El-Yaniv. 2018. Deep anomaly detection using geometric transformations. In *NIPS*.

David Guthrie. 2008. *Unsupervised Detection of Anomalous Text*. Ph.D. thesis, University of Sheffield.

Dan Hendrycks, Mantas Mazeika, and Thomas Dietterich. 2019. Deep anomaly detection with outlier exposure. *ICLR*.

Felix Hill, Kyunghyun Cho, and Anna Korhonen. 2016. Learning distributed representations of sentences from unlabelled data. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1367–1377, San Diego, California. Association for Computational Linguistics.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2017. Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 427–431, Valencia, Spain. Association for Computational Linguistics.

Ramakrishnan Kannan, Hyenkyun Woo, Charu C Aggarwal, and Haesun Park. 2017. Outlier detection for text data. In *SDM*, pages 489–497.

Yoon Kim. 2014. *Convolutional neural networks for sentence classification*. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar. Association for Computational Linguistics.

D. Kingma and J. Ba. 2014. Adam: A Method for Stochastic Optimization. *arXiv:1412.6980*. 

4069
Raksha Kumaraswamy, Anurag Wazalwar, Tushar Khot, Jude W Shavlik, and Srijita Natarajan. 2015. Anomaly detection in text: The value of domain knowledge. In FLAIRS Conference, pages 225–228.

Sebastian Lapuschkin, Stephan Waldchen, Alexander Binder, Grégoire Montavon, Wojciech Samek, and Klaus-Robert Müller. 2019. Unmasking clever hands predictors and assessing what machines really learn. Nature communications, 10(1):1096.

Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In ICML, pages 1188–1196.

Ji Young Lee and Franck Dernoncourt. 2016. Sequential short-text classification with recurrent and convolutional neural networks. In Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 515–520, San Diego, California. Association for Computational Linguistics.

Zhouhan Lin, Minwei Feng, Cícero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. 2017. A structured self-attentive sentence embedding. In ICLR.

Bing Liu, Wee Sun Lee, Philip S Yu, and Xiaoli Li. 2002. Partially supervised classification of text documents. In ICML, volume 2, pages 387–394.

Amogh Mahapatra, Nisheeth Srivastava, and Jaideep Srivastava. 2012. Contextual anomaly detection in text data. Algorithms, 5(4):469–489.

Pankaj Malhotra, Anusha Ramakrishnan, Gaurangi Anand, Lovekesh Vig, Puneet Agarwal, and Gautam Shroff. 2016. LSTM-based encoder-decoder for multi-sensor anomaly detection. In Anomaly Detection Workshop at 33rd International Conference on Machine Learning.

Pankaj Malhotra, Lovekesh Vig, Gautam Shroff, and Puneet Agarwal. 2015. Long short term memory networks for anomaly detection in time series. In European Symposium on Artificial Neural Networks.

Larry M Manevitz and Malik Yousef. 2001. One-class svm for document classification. JMLR, 2(Dec):139–154.

Larry M Manevitz and Malik Yousef. 2007. One-class document classification via neural networks. Neurocomputing, 70(7-9):1466–1481.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositional-ity. In NIPS, pages 3329–3332.

Mary M Moya, Mark W Koch, and Larry D Hostetler. 1993. One-class classifier networks for target recognition applications. In Proceedings World Congress on Neural Networks, pages 797–801.

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.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Marco AF Pimentel, David A Clifton, Lei Clifton, and Lionel Tarassenko. 2014. A review of novelty detection. Signal Processing, 99:215–249.

Andreas Rücklé, Steffen Eger, Maxime Peyrard, and Iryna Gurevych. 2018. Concatenated p-mean word embeddings as universal cross-lingual sentence representations. arXiv:1803.01400.

Lukas Ruff, Robert A. Vandermeulen, Nico Görnitz, Lucas Deecke, Shaib A. Siddiqui, Alexander Binder, Emmanuel Müller, and Marius Kloft. 2018. Deep one-class classification. In ICML, volume 80, pages 4390–4399.

M. Sakurada and T. Yairi. 2014. Anomaly detection using autoencoders with nonlinear dimensionality reduction. In Proceedings of the 2nd MLSDA Workshop, page 4.

T. Schlegl, P. Seeböck, S. M. Waldstein, U. Schmidt-Erfurth, and G. Langs. 2017. Unsupervised Anomaly Detection with Generative Adversarial Networks to Guide Marker Discovery. In IPMI, pages 146–157.

B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson. 2001. Estimating the Support of a High-Dimensional Distribution. Neural computation, 13(7):1443–1471.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositional-ity over a sentiment treebank. In Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

D. M. J. Tax and R. P. W. Duin. 2004. Support Vector Data Description. Machine learning, 54(1):45–66.
Yashen Wang, Heyan Huang, Chong Feng, Qiang Zhou, Jiahui Gu, and Xiong Gao. 2016. CSE: Conceptual sentence embeddings based on attention model. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 505–515, Berlin, Germany. Association for Computational Linguistics.

John Wieting, Mohit Bansal, Kevin Gimpel, and Karen Livescu. 2016. Towards universal paraphrastic sentence embeddings. In ICLR.

D. Xu, E. Ricci, Y. Yan, J. Song, and N. Sebe. 2015. Learning Deep Representations of Appearance and Motion for Anomalous Event Detection. In BMVC, pages 8.1–8.12.