Exploring Textual Out-Of-Distribution Detection: from simplistic supervised to advanced self-supervised techniques

Camille Langlois
Ensae
camille.langlois@ensae.fr

Joé Vincent-Galtié
Ensae
joe.vincent-galtie@ensae.com

Abstract

This essay delves into the issue of Textual Out-Of-Distribution (OOD) detection, which refers to the capability of machine learning models to recognize data samples that significantly deviate from their training data distribution. In Natural Language Processing (NLP) applications, Textual OOD detection is critical to ensuring the robustness and dependability of production systems. This study investigates the effectiveness of various methods for OOD detection in NLP, utilizing a transformer-based language model and different datasets with varying degrees of similarity to the training data. Our findings demonstrate that both the Mahalanobis-based score utilizing the last layer representation and the Cosine Projection score utilizing the average latent representation outperform the other scores in terms of AUROC. However, the supervised approach did not perform as well. Code is available on github [1]

1 Problem Framing

Increasing the use of black-box machine learning models comes with various critical safety issues, among which we can mention the Textual Out-of-Distribution (OOD) detection [2]. The goal of OOD detection is to identify instances that are significantly different from the distribution of the training data, which can be caused by various factors such as errors, noise, or deliberate attempts to deceive the system [16, 17]. However, distinguishing OOD from in-distribution (ID) examples is difficult for modern deep neural architectures [6, 7, 11], as these models transform incoming data into latent representations that make reliable information extraction challenging. In the present paper, we adress the issue of OOD detection on classifiers for textual data, we will more particularly focus on models based on Transformer architectures [20].

The existing methods adressing the OOD detection issue can be categorized based on their positioning with respect to the network, including those that use incoming data [10, 19], robust constraints during training [13], and post-processing methods. The post-processing methods are considered the most promising because they do not require retraining and can be used on any pre-trained model. These methods include softmax-based tools that compute a confidence score based on predicted probabilities and threshold [12], projections of the pre-softmax layer [14], and the Mahalanobis distance between a test sample and the in-distribution law estimated through accessible training data points [15]. Other approaches based on the concept of data depth have arisen to overcome the drawbacks of distance-based scores, such as using the Integrated Rank-Weighted depth [4].

In this essay, we present an experimentation on the performance of different OOD detection techniques on a benchmark dataset of text classification tasks [5]. We first propose a simplistic supervised method relying on a XGBoost [3] model. We also evaluate the effectiveness of various methods, including scores based on Mahalanobis-distance computation and Cosine Projection, in detecting OOD examples in both in-domain and out-of-domain datasets. Our results demonstrate the strengths and limitations of different approaches and provide insights for future research on improving the reliability of OOD detection in NLP tasks.

[1] https://github.com/joevincentgaltie/OOD_Detection_ENSAE.git

2 Experiments Protocol

In this section we will introduce the chosen benchmark, the pretrained encoders and the baseline methods that we experimented in order to compare the results.

2.1 Datasets selection

During the experiments, we are going to consider three different datasets. The models will be trained on one of the datasets, which will then correspond to the in-distribution data. In this case, the three datasets chosen are SST2[2], IMDB[3] and RTE[4]. SST2 (see Table 1) is a sentiment analysis dataset that contains movie reviews with binary labels indicating positive or negative sentiment. On the other hand, IMDB (see Table 2) is also a sentiment analysis dataset but it contains reviews of a wider range of products, such as books, electronics, and home appliances. RTE (see Table 3) consists of pairs of sentences, where the task is to determine whether one sentence entails, or contradicts with respect to the other. In our case SST2 represents the ID data. Since IMDB and RTE are not part of the distribution on which the models were trained on, it serves as the OOD datasets in this experiment.

# of samples	67 349
Average sentence length	19.8
# of classes	2
Language	English

Table 1: Features of the SST2 dataset.

# of samples	50 000
Average sentence length	231.73
# of classes	2
Language	English

Table 2: Features of the IMDB dataset.

2.2 Pretrained model selection

The experiments regarding the scorers have been done regarding a pretrained encoder. We apply the various scorers on the BERT [9] model. The selected model has been pretrained and fine-tuned on SST2 dataset and will be used to extract features from the input text for both IN and OOD datasets. We have chosen this model because it is among the most widely used and effective model for NLP tasks.

[2] https://huggingface.co/datasets/sst2
[3] https://huggingface.co/datasets/imdb
[4] https://huggingface.co/datasets/SetFit/rte

# of samples	2 490
Average sentence length	68.6
# of classes	2
Language	Multi.

Table 3: Features of the RTE dataset.

2.3 Simplistic supervised OOD Detection

Our supervised approach of the OOD detection is simplistically framed. Using PyTorch and the BERT [9] model already fine-tuned on the SST2 dataset, we iterated on batches of SST2 samples in order to retrieve hidden states corresponding to each of the 13 hidden layers.

For a batch of 8 sentences, we therefore have for each layer 8 matrix of dimension $T \times d$ where T is the number of tokens per sentences and d is the embedding dimension.

We introduce the following notation:

- $\forall b \in \{1, ..., B\}$ where B is the number of batches
- $\forall l \in \{1, ..., L\}$ where L is the number of layers (13)
- $\forall x_{i,b}$ for $i \in [1, 8]$, 8 being the size of the batch
- $H^{l=1}_{x_{i,b}} = (h^{l=1}_{i,j}) \in M_{T \times 768}$
- $\bar{x}_{i,b} = (\frac{1}{13}\sum_{l=1}^{13} h^{l}_{1,1}, ..., \frac{1}{13}\sum_{l=1}^{13} h^{l}_{1,768})$
- $X_{SST2} = (\bar{x}_{1,1}, ..., \bar{x}_{8,1}, \bar{x}_{1,2}, ...)$

The same steps are processed for IMDB (out-ds) and RTE (very-out) datasets.

We then concatenated X_{SST2} and X_{IMDB}, and X_{SST2} and X_{RTE}, completed by labels, 0 if in-ds, 1 if out or very-out ds.

We apply a classification supervised algorithm. In particular we applied a XGBoost [3], one of the current most performant classification algorithm.

2.4 Self-supervised OOD Detection : Scorers

For our experiments we considered two different methods:

- The Mahalanobis based score [8]: this score measures the distance between a given input and the distribution of ID examples in the latent space of a pretrained language model. The Mahalanobis distance takes into account the covariance matrix of the ID examples, and thus can better capture the distribution of the data than other distance metrics like Euclidean distance or Cosine distance. It can be computed as:

$$\mathrm{d}_{Mah}(\mathbf{x}) = \sqrt{((f(\mathbf{x}) - \mu)^T S^{-1} (f(\mathbf{x}) - \mu))}$$

  where f(**x**) represents the latent representation for a given input example **x**, μ is the mean of the ID training data, and S is the covariance matrix of the ID training data. In our study we will consider the latent representation to be either the vector of activations in the last hidden layer of the neural network and the average representation of all the layers.

- The Cosine Projection based score: this is a commonly used metric to compare the similarity between a given input and the distribution of ID examples using the Cosine similarity. Given two vectors u and v, their Cosine similarity score is defined as:

$$cosine\_sim(u, v) = \frac{u \cdot v}{||u||||v||}$$

  More precisely, the cosine similarity score can be used to compare the similarity between the latent representations of a given text sample and the ID examples used during the training of a language model. Specifically, given a test sample **x** and a set of ID examples X_{in}, we can compute the Cosine similarity between the latent representation of **x**, denoted as $f(\mathbf{x})$, and the average latent representation of the ID examples, denoted as $f(X_{in})$:

$$cos\_score(\mathbf{x}) = cosine\_sim(f(\mathbf{x}), f(X_{in}))$$

  Again, we will on one side consider only the latent representation of the last layer, and on the other side the average of the representations of each layer.

In both cases, a threshold is used to classify the test sample as either ID or OOD. If the score is below the threshold, the sample is classified as ID, otherwise it is classified as OOD. The threshold can be set using a validation set or a predefined value.

We can also note that both methods require access to ID training data to estimate the mean and covariance matrix for the Mahalanobis-based score, and the mean for the Cosine Projection score.

2.5 Last vs average of every embedding layer

Following [4], we tested the scorer on the embeddings of the last layer and on the average embedding of all hidden layers.
To proceed so is a way to distinguish whether the intuition that more information comes from agregating results of each layer.
Hence, for the second case, the input x is such that $x \in R^{d=768}$ and $x = \sum_{l=1}^{L} x_1^l$ where $x_1^l \in R^{d=768}$ corresponds to the l-th hidden state of the first token

2.6 Evaluation metrics

There exist several ways to measure the effectiveness of an OOD method. Here will focus on the Area Under the Receiver Operating Characteristic curve (AUROC) metric [1], which is a commonly used evaluation metric to assess the performance of a model in distinguishing between ID and OOD samples.

The ROC curve refers to a plot of the True Positive Rate (TPR) against the False Positive Rate (FPR) for different threshold values. In the context of OOD detection, the TPR represents the proportion of correctly identified OOD samples, while the FPR represents the proportion of incorrectly identified ID samples as OOD.

AUROC corresponds to the area under the ROC curve, ranging from 0.0 to 1.0. An AUROC score of 1.0 indicates perfect performance, while a score of 0.5 indicates random guessing. A score below 0.5 indicates poor performance, which means the model is worse than random guessing.

3 Results

3.1 Simplisitic supervised approach

This approach did not give convincing results. In fact, the XGBoost gave a too perfect accuracy for classifying SST2 vs IMDB and SST2 vs RTE to make this approach worth it.

After a deeper study of the results, this perfect accuracy was induced by a dimension of the mean embedding, the 135th, that was always negative for SST2 and always positive for IMBD.

We did not figure out yet why this happened. However it is worth noticing that this approach has obvious limits :

- This supposes having access to numerous diversed OOD sentences and implies a too heavy labelling work on real issues.

3.2 Self-supervised approach

The experimentation performed on the methods and datasets mentioned above enable to have some insights regarding which scorers are the most adapted taking into account the model used and the datasets. We can for instance see in Figure 1 the distribution of the scorers for each dataset. These graphs allow us to notice that the scorers distributions of in-ds and (very) out-ds are more clearly distinguishable for the Mahanalobis-based scorer that uses the last layer, and the Cosine Projection scorer that uses the average latent representation.

We can draw the same observations if we look at the AUROC given in the Figure 2. Again, the AUROC is higher for the Mahanalobis-based scorer that uses the last layer, and the Cosine Projection scorer that uses the average latent representation than for the other scores.

4 Discussion/Conclusion

4.1 Computational issues

To do this work we encountered several computational issues that impeded our progress. Unfortunately, these issues caused delays in the project timeline and impacted the scope of our experimentation.

4.2 Elements to be further explored

With more time during this project, we would have liked to bring the following experiments:

- It would have been interesting to test other detectors, especially TRUSTED introduced by [4].
- Also, as mentionned in [4], it is important to test the different scorers on various models to fully evaluate their performance. We would have liked to carry out our experiments on the DISTILBERT [18] model.

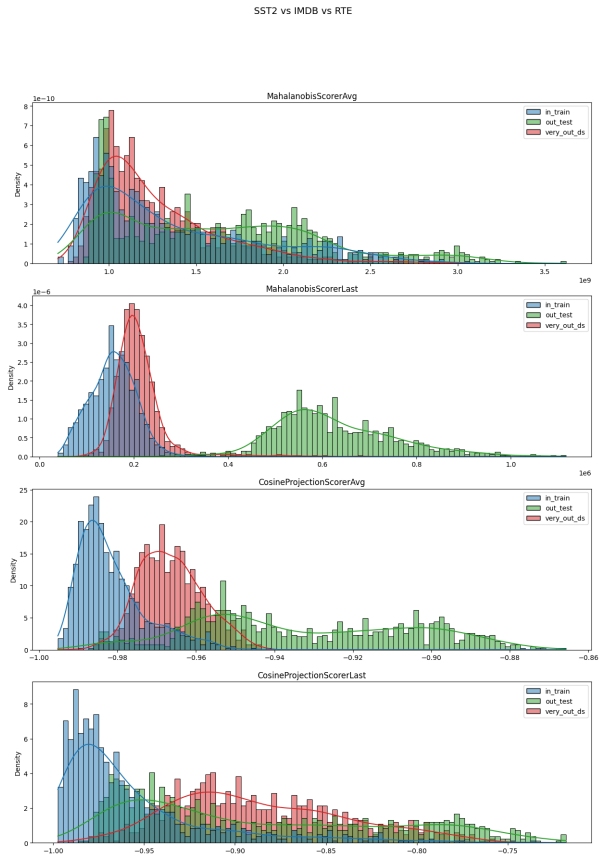

Figure 1: Scorers distribution on datasets (in_ds[SST2],out_ds[IMDB],very_out[RTE])

4.3 Pairing datasets as IN/OUT

The unsatisfying results obtained on last layer scorers might come from the pairing of SST2/IMDB as these are close semantic datasets that requires high-end OOD detectors using information of many hidden-layers and not only the last one.

4.4 Conclusion

In conclusion, our experimentation focused on the task of out-of-distribution (OOD) detection for text classification. We used the SST2, IMDB and RTE datasets to evaluate the performance of different OOD detection methods, namely Mahalanobis-based and Cosine Projection-based scores. Our results showed that both Mahalanobis-based and Cosine Projection-based scores are effective in OOD detection for text classification. Specifically, the Mahalanobis-based score performed best using only the last layer for the latent representation, and the Cosine Projection-based score performed best using the average latent representation. However, our supervised approach

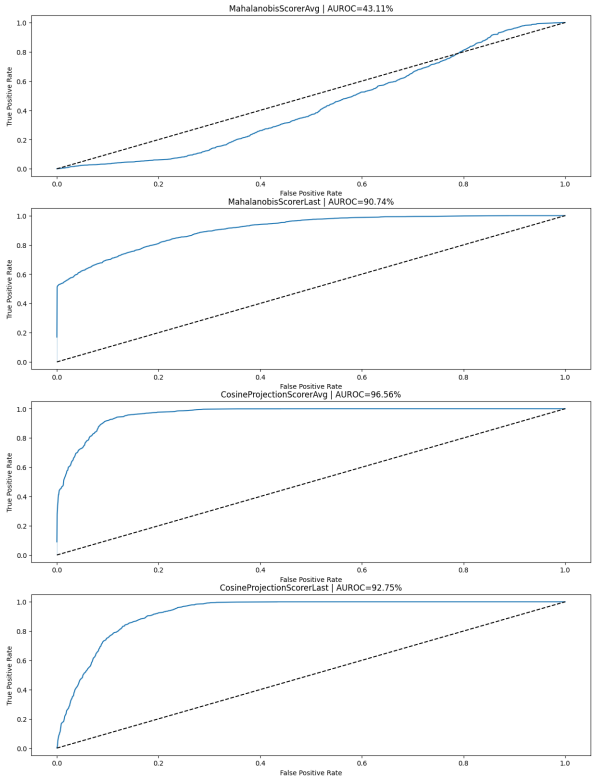

Figure 2: AUROC for each scorer.

did not perform as well as expected. Despite some computational issues, our findings suggest that OOD detection methods are promising for text classification tasks and warrant further investigation.

References

[1] A. P. Bradley. The use of the area under the roc curve in the evaluation of machine learning algorithms. *Pattern recognition*, 30(7):1145–1159, 1997.

[2] S. Bulusu, B. Kailkhura, B. Li, P. K. Varshney, and D. Song. Anomalous example detection in deep learning: A survey. *IEEE Access*, 8:132330–132347, 2020.

[3] T. Chen, T. He, M. Benesty, V. Khotilovich, Y. Tang, H. Cho, K. Chen, R. Mitchell, I. Cano, T. Zhou, et al. Xgboost: extreme gradient boosting. *R package version 0.4-2*, 1(4):1–4, 2015.

[4] P. Colombo, E. D. C. Gomes, G. Staerman, N. Noiry, and P. Piantanida. Beyond mahalanobis distance for textual ood detection. In *NeurIPS 2022*, 2022.

[5] P. Colombo, N. Noiry, E. Irurozki, and S. Clémençon. What are the best systems? new perspectives on nlp benchmarking. *NeurIPS 2022*, 2022.

[6] M. Darrin, P. Piantanida, and P. Colombo. Rainproof: An umbrella to shield text generators from out-of-distribution data. *arXiv preprint arXiv:2212.09171*, 2023.

[7] M. Darrin, G. Staerman, E. D. C. Gomes, J. C. Cheung, P. Piantanida, and P. Colombo. Unsupervised layer-wise score aggregation for textual ood detection. *arXiv preprint arXiv:2302.09852*, 2023.

[8] R. De Maesschalck, D. Jouan-Rimbaud, and D. L. Massart. The mahalanobis distance. *Chemometrics and intelligent laboratory systems*, 50(1):1–18, 2000.

[9] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[10] V. Gangal, A. Arora, A. Einolghozati, and S. Gupta. Likelihood ratios and generative classifiers for unsupervised out-of-domain detection in task oriented dialog. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 7764–7771, 2020.

[11] E. D. C. Gomes, P. Colombo, G. Staerman, N. Noiry, and P. Piantanida. A functional perspective on multi-layer out-of-distribution detection.

[12] D. Hendrycks and K. Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *arXiv preprint arXiv:1610.02136*, 2016.

[13] D. Hendrycks, M. Mazeika, and T. Dietterich. Deep anomaly detection with outlier exposure. *arXiv preprint arXiv:1812.04606*, 2018.

[14] W. Liu, X. Wang, J. Owens, and Y. Li. Energy-based out-of-distribution detection. *Advances in neural information processing systems*, 33:21464–21475, 2020.

[15] P. C. Mahalanobis. On the generalised distance in statistics. In *Proceedings of the national Institute of Science of India*, volume 12, pages 49–55, 1936.

[16] M. Picot, N. Noiry, P. Piantanida, and P. Colombo. Adversarial attack detection under realistic constraints. 2023.

[17] M. Picot, G. Staerman, F. Granese, N. Noiry, F. Messina, P. Piantanida, and P. Colombo. A simple unsupervised data depth-based method to detect adversarial images. 2023.

[18] V. Sanh, L. Debut, J. Chaumond, and T. Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.

[19] G. Staerman, P. Mozharovskyi, P. Colombo, S. Clémençon, and F. d'Alché Buc. A pseudo-metric between probability distributions based on depth-trimmed regions. *arXiv e-prints*, pages arXiv–2103, 2021.

[20] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.