Holistic Sentence Embeddings for Better Out-of-Distribution Detection

Sishuo Chen¹, Xiaohan Bi¹, Rundong Gao¹, Xu Sun²
¹Center for Data Science, Peking University
²MOE Key Laboratory of Computational Linguistics, School of Computer Science, Peking University
{chensishuo,xusun}@pku.edu.cn {bxh,gaord20}@stu.pku.edu.cn

Abstract

Detecting out-of-distribution (OOD) instances is significant for the safe deployment of NLP models. Among recent textual OOD detection works based on pretrained language models (PLMs), distance-based methods have shown superior performance. However, they estimate sample distance scores in the last-layer CLS embedding space and thus do not make full use of linguistic information underlying in PLMs. To address the issue, we propose to boost OOD detection by deriving more holistic sentence embeddings. On the basis of the observations that token averaging and layer combination contribute to improving OOD detection, we propose a simple embedding approach named Avg-Avg, which averages all token representations from each intermediate layer as the sentence embedding and significantly surpasses the state-of-the-art on a comprehensive suite of benchmarks by a 9.33% FAR95 margin. Furthermore, our analysis demonstrates that it indeed helps preserve general linguistic knowledge in fine-tuned PLMs and substantially benefits detecting background shifts. The simple yet effective embedding method can be applied to fine-tuned PLMs with negligible extra costs, providing a free gain in OOD detection. Our code is available at https://github.com/lancopku/Avg-Avg.

1 Introduction

Pretrained language models have achieved remarkable performance on various NLP tasks under the assumption that the train and test samples are drawn from the same distribution (Wang et al., 2019). However, in real-life applications such as dialogue systems and clinical text processing, it is inevitable for models to make predictions on out-of-distribution (OOD) samples, which may result in fatally unreasonable predictions (Hendrycks et al., 2020). Therefore, it is crucial for fine-tuned PLMs to automatically detect OOD inputs.

Among recent works on textual OOD detection, distance-based methods have received much attention due to their superior performance (Podolskiy et al., 2021; Zhou et al., 2021). They calculate the sample distance to the training-data distribution as the uncertainty measure for OOD detection. In these approaches, the distance scores are usually calculated in the space of the last-layer CLS vectors (i.e., the inputs to the classification head) produced by fine-tuned PLMs. As known, the CLS embedding space is optimized for the in-distribution classification task during fine-tuning, thus not necessarily optimal for OOD detection.

In this paper, we investigate how to derive sentence embeddings suitable for OOD detection from fine-tuned PLMs. Motivated by the token averaging and layer combination techniques proposed to enhance unsupervised sentence embeddings (Su et al., 2021; Huang et al., 2021b), we apply them to OOD detection and make two intriguing empirical findings: (1) averaging all token representations outperforms the standard practice only using the CLS vector; (2) combining token representations from all intermediate layers brings further improvements. These observations lead to an extremely simple yet effective pooling technique: averaging all token representations in each intermediate layer as the sentence embedding for OOD detection.

We name the all-layer-all-token pooling technique Avg-Avg and demonstrate that it consistently uplifts the OOD detection performance of BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019) models on a comprehensive suite of textual OOD detection benchmarks. Further investigations into the rationales behind the improvement show that Avg-Avg effectively helps reserve general linguistic information in the feature space and benefits detecting background shifts. In summary, our proposal serves as a plug-and-play post-processing technique to improve the capability of fine-tuned
PLMs to detect OOD instances and reveals that it is a promising direction to boost textual OOD detection via deriving more holistic representations.

2 Avg-Avg: Holistic Sentence Embedding for Better OOD Detection

2.1 Preliminaries

Modern pretrained language models have been developed based on the Transformer (Vaswani et al., 2017) architecture. Given a sentence \( S = \{t_1, t_2, \ldots, t_n\} \) as the input, an L-layer Transformer-based PLM yields a series of hidden vectors \( \mathbb{H} = \{H_0, H_1, \ldots, H_L\} \), where \( H_i = [h_i^1, h_i^2, \ldots, h_i^n] \) \((0 < i \leq L)\) are the embedding vectors for each token in \( S \) in the \( i \)-th Transformer layer and \( H_0 \) denotes the static token embeddings.

2.2 Methodology

In the pretraining-finetuning paradigm, the CLS token is usually put at the beginning of \( S \), and the corresponding vector produced by the last Transformer layer \( h_1^n \) is fed into the classification head for fine-tuning. In existing works, the CLS vector \( h_1^n \) is regarded as the sentence representation, and OOD detection is conducted in the corresponding embedding space (Podolskiy et al., 2021; Zhou et al., 2021). Such a practice does not fully exploit linguistic information contained in \( \mathbb{H} \). Consequently, we resort to two pooling strategies to derive more holistic sentence representations:

- **Intra-Layer Token Averaging**: For the \( i \)-th layer, we average hidden vectors for all tokens as the pooled representation \( P_i \), i.e., \( P_i = \frac{1}{n} \sum_{j=1}^{n} h_i^j \), to replace the default \( P_1 = h_1^n \).

- **Inter-Layer Combination**: For given intermediate pooled representations \( P_1, P_2, \ldots, P_L \), we perform layer combination to obtain the final pooled sentence representation \( P \) for OOD detection: \( P = \frac{1}{|M|} \sum_{i \in M} P_i \), where \( M \subseteq \{1, 2, \ldots, L\} \) denotes the subset of intermediate layers for combination.

In our embedding approach Avg-Avg, token averaging is performed for intra-layer pooling; all layers are chosen for layer combination, in other words, \( M = \{1, 2, \ldots, L\} \). Table 1 shows the rationality of our choice: for a RoBERTa-based model fine-tuned on the SST-2 sentiment analysis dataset, Avg-Avg significantly outperforms other pooling strategies for detecting 20 Newsgroup (20NG) samples as OOD data, including the default last-layer CLS pooling and the first-last-avg pooling used for unsupervised sentence embedding (Su et al., 2021).

### Table 1: The performance of different pooling strategies on the SST-2 v.s. 20NG benchmark. Mahalanobis distance (Lee et al., 2018) is the OOD detection method. Avg denotes token average pooling and L12 denotes the 12th layer (the last) of the RoBERTa model. These results are exploratory and the superiority of Avg-Avg will be further confirmed by following experiments.

<table>
<thead>
<tr>
<th>Intra-Layer</th>
<th>Inter-Layer</th>
<th>AUROC%</th>
<th>Remark</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLS</td>
<td>L12</td>
<td>90.48</td>
<td>Default</td>
</tr>
<tr>
<td>Avg</td>
<td>L12</td>
<td>93.17</td>
<td></td>
</tr>
<tr>
<td>CLS</td>
<td>All Layers</td>
<td>94.34</td>
<td></td>
</tr>
<tr>
<td>Avg</td>
<td>L1+L12</td>
<td>98.65</td>
<td>first-last-avg</td>
</tr>
<tr>
<td>Avg</td>
<td>All Layers</td>
<td>99.99</td>
<td>Avg-Avg (Ours)</td>
</tr>
</tbody>
</table>

3 Experiments

3.1 Experimental Setup

**Benchmarks** Following Zhou et al. (2021), we choose four datasets corresponding to three tasks as the in-distribution (ID) datasets: SST-2 (Socher et al., 2013) and IMDB (Maas et al., 2011) for sentiment analysis, TREC-10 (Li and Roth, 2002) for question classification, and 20 Newsgroups (Lang, 1995) for topic classification. Among the four, any pair of datasets coming from different tasks is regarded as an ID-OOD. Besides, we use four additional datasets as OOD test data for each ID dataset: WMT-16 (Bojar et al., 2016), Multi30k (Elliott et al., 2016), RTE (Dagan et al., 2005), and SNLI (Bowman et al., 2015). More details of these datasets can be found in Appendix A.1.

**Model Configuration** We build text classifiers by fine-tuning the RoBERTa-base model (Liu et al., 2019) (110M parameters) in main experiments. Our implementation is based on Hugging Face’s Transformers library (Wolf et al., 2020). Please refer to Appendix B for more details.

**Evaluation Protocol** For OOD detection performance, we report AUROC and FAR95 following Zhou et al. (2021). Higher AUROC and lower FAR95 values indicate better OOD detection performance (specific definitions in Appendix C).

**Baselines for Comparison** We reimplement a series of existing OOD detection methods for comparison: MSP (Hendrycks and Gimpel, 2017), Energy Score (Liu et al., 2020), LOF (Lin and Xu, 2019), Mahalanobis distance (MD for short) (Lee et al., 2018), Energy Score (Liu et al., 2020), LOF (Lin and Xu, 2019), Mahalanobis distance (MD for short) (Lee et al., 2018).
Table 2: The AUROC / FAR95 results of previous OOD detection methods and ours on four benchmarks. ↑ indicates larger is better and ↓ indicates lower is better. For each ID dataset, we report the macro average of AUROC / FAR95 values on all corresponding OOD datasets. All values are percentages averaged over five times with different random seeds, and the best results are highlighted in **bold.** $L_{\text{cl}}$ and $L_{\text{margin}}$ denote the contrastive and margin-based auxiliary targets proposed by Zhou et al. (2021), respectively.

<table>
<thead>
<tr>
<th>AUROC↑ / FAR95↓</th>
<th>SST-2</th>
<th>IMDB</th>
<th>TREC-10</th>
<th>20NG</th>
<th>Avg.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Baselines</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MSP (Hendrycks and Gimpel, 2017)</td>
<td>88.10 / 70.00</td>
<td>96.36 / 22.43</td>
<td>94.28 / 24.15</td>
<td>87.96 / 49.95</td>
<td>91.68 / 41.63</td>
</tr>
<tr>
<td>LOF (Lin and Xu, 2019)</td>
<td>78.63 / 66.29</td>
<td>89.36 / 54.19</td>
<td>96.37 / 22.11</td>
<td>92.56 / 36.81</td>
<td>89.23 / 44.85</td>
</tr>
<tr>
<td>Energy (Liu et al., 2020)</td>
<td>87.47 / 72.43</td>
<td>95.83 / 23.95</td>
<td>95.50 / 19.68</td>
<td>90.37 / 32.31</td>
<td>92.29 / 37.09</td>
</tr>
<tr>
<td>MD Baseline (Podol斯基 et al., 2021)</td>
<td>91.88 / 48.82</td>
<td>99.19 / 2.86</td>
<td>99.12 / 2.25</td>
<td>96.75 / 15.75</td>
<td>96.74 / 17.42</td>
</tr>
<tr>
<td>MD + $L_{\text{cl}}$ (Zhou et al., 2021)</td>
<td>92.16 / 49.04</td>
<td>98.86 / 4.08</td>
<td>98.59 / 4.96</td>
<td>95.47 / 21.56</td>
<td>96.27 / 19.91</td>
</tr>
<tr>
<td>MD + $L_{\text{margin}}$ (Zhou et al., 2021)</td>
<td>95.35 / 29.43</td>
<td><strong>99.93 / 0.15</strong></td>
<td>99.36 / 1.72</td>
<td>96.51 / 18.51</td>
<td>97.79 / 12.45</td>
</tr>
<tr>
<td><strong>Ours</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>token=AVG, layer=L12</td>
<td>93.14 / 41.24</td>
<td>99.67 / 1.09</td>
<td>98.29 / 2.88</td>
<td>97.07 / 14.07</td>
<td>97.04 / 14.82</td>
</tr>
<tr>
<td>token=CLS, layer=ALL</td>
<td>93.93 / 34.73</td>
<td>99.53 / 1.35</td>
<td>99.33 / 1.29</td>
<td>96.73 / 15.67</td>
<td>97.38 / 13.26</td>
</tr>
<tr>
<td>token=AVG, layer=ALL (Avg-Avg)</td>
<td><strong>97.75 / 10.67</strong></td>
<td><strong>99.87 / 0.21</strong></td>
<td><strong>99.66 / 0.23</strong></td>
<td><strong>99.70 / 1.35</strong></td>
<td><strong>99.25 / 3.12</strong></td>
</tr>
</tbody>
</table>

Table 3: The improvements brought by Avg-Avg compared to the MD baseline (Podol斯基 et al., 2021) for different PLMs. AUROC values are reported (the number in the bracket is the improvement).

<table>
<thead>
<tr>
<th>AUROC↑</th>
<th>SST-2</th>
<th>IMDB</th>
<th>TREC-10</th>
<th>20NG</th>
<th>Avg.</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALL-BERT-base</td>
<td>95.41 (±8.62)</td>
<td>99.51 (±1.10)</td>
<td>98.82 (±0.98)</td>
<td>99.55 (±7.02)</td>
<td>98.51 (±6.88)</td>
</tr>
<tr>
<td>DistilBERTs-base</td>
<td>97.80 (±3.17)</td>
<td>99.87 (±1.10)</td>
<td>99.17 (±0.97)</td>
<td>99.94 (±4.15)</td>
<td>99.20 (±5.56)</td>
</tr>
<tr>
<td>RoBERTa-base</td>
<td>97.78 (±2.22)</td>
<td>99.51 (±0.45)</td>
<td>98.81 (±0.42)</td>
<td>99.79 (±1.60)</td>
<td>98.85 (±0.95)</td>
</tr>
<tr>
<td>RoBERTa-base</td>
<td>97.75 (±5.87)</td>
<td>99.87 (±0.68)</td>
<td>99.66 (±5.54)</td>
<td>99.70 (±2.33)</td>
<td>99.25 (±2.51)</td>
</tr>
<tr>
<td>RoBERTa-base</td>
<td>97.48 (±5.13)</td>
<td>99.94 (±1.24)</td>
<td>99.62 (±0.78)</td>
<td>99.97 (±2.88)</td>
<td>99.21 (±2.79)</td>
</tr>
</tbody>
</table>

Figure 1: Maximum AUROC (averaged over 6 OOD datasets) values corresponding to sentence embeddings from a RoBERTa model fine-tuned on SST-2 with different numbers of combining layers. The maximum values are results searched on the test data. Token averaging is performed for intra-layer pooling.

3.2 Overall Results

Table 2 gives main results. Except the contrastive-based tuning method (Zhou et al., 2021), all methods use the same model vanilla fine-tuned on the ID training set. Our methods use the Mahalanobis distance to obtain OOD scores, following Podol斯基 et al. (2021) (the only difference lies in the embedding space). We find that compared to the baseline calculating MD in the last-layer CLS embedding space, both token averaging and layer combination bring improvements on almost all benchmarks. When the two techniques are combined, namely Avg-Avg is applied, the performance continues to grow and exceeds the previous state-of-the-art (Zhou et al., 2021) that needs extra contrastive targets in the fine-tuning stage, by a considerable margin of 9.33% FAR95 averaged on four benchmarks. Further experiments on other PLM backbones also substantiate the enhancement brought by our method, supported by the results in Table 3.

3.3 Analysis

The Impact of Layer Choice To verify the rationality of choosing all intermediate layers for inter-layer combination, we show the maximum AUROC values corresponding to different numbers of intermediate layers to derive sentence embeddings in Figure 1. As the number of layers grows, the AUROC metric first increases and then remains relatively stable when more than four layers are chosen. Notably, the peak appears when 7 layers are combined, only 0.3% higher than our Avg-Avg. Since searching for the best combination of layers is infeasible due to the unavailability of OOD data, using all layers is a sensible choice.

Probing Analysis Given that intermediate layers of PLMs contain a rich hierarchy of linguistic information (Jawahar et al., 2019), a plausible explanation of the performance lift is that Avg-Avg leads to an embedding space containing more gen-
Intra-Layer Inter-Layer Surface Syntactic Semantic Average SentLen WC TredDepth TopConst BShift Tense SubjNum ObjNum SOMO CoordInv

<table>
<thead>
<tr>
<th>Dominant Shift</th>
<th>ID</th>
<th>OOD</th>
<th>AUROC Baseline</th>
<th>Ours</th>
</tr>
</thead>
<tbody>
<tr>
<td>Background</td>
<td>SST-2</td>
<td>IMDB</td>
<td>69.86</td>
<td>97.57 (+27.70)</td>
</tr>
<tr>
<td></td>
<td>CR</td>
<td></td>
<td>75.46</td>
<td>82.26 (+6.80)</td>
</tr>
<tr>
<td>Semantic</td>
<td>News Top-5</td>
<td>News Rest</td>
<td>83.41</td>
<td>83.83 (+0.42)</td>
</tr>
<tr>
<td></td>
<td>CLINC</td>
<td>CLINC OOD</td>
<td>97.58</td>
<td>97.88 (+0.30)</td>
</tr>
</tbody>
</table>

Table 4: Probing task performance for representations corresponding to different pooling strategies. All values are percentages averaged over five RoBERTa models fine-tuned with different random seeds.

Table 5: Performance (AUROC) on different distribution shifts, corresponding to the MD baseline and our proposed Avg-Avg. All values are percentages averaged over five different random seeds.

eeral linguistic information, where ID and OOD data are more sharply separated. To verify this, we evaluate the sentence embeddings produced by the RoBERTa model fine-tuned on SST-2 corresponding to different pooling strategies on the probing tasks proposed by Conneau et al. (2018) (details in Appendix A.3). As shown in Table 4, our proposed method consistently raises the probing accuracies of surface, syntactic, and semantic level probing tasks, suggesting that we obtain more holistic embeddings by integrating intermediate hidden states.

Detecting Different Kinds of Shifts OOD texts can be categorized by whether they exhibit a background shift or a semantic shift (Arora et al., 2021). In previous main experiments, ID and OOD data come from different tasks and both kinds of shifts exist. To explore the source of the performance growth, we conduct ablation experiments by evaluating our method in settings where background or semantic shifts dominate. For the semantic shift setting, we use the News Category (Misra, 2018) and CLINC (Larson et al., 2019a) datasets (ID and OOD parts share the same background distribution, but belong to different classes); for the background shift setting, we regard SST-2 as ID and IMDB, Customer Reviews (CR for short) (Hu and Liu, 2004) as OOD (they all belong to the sentiment analysis task but differ in background features, e.g., the length and style). Refer to Appendix A.2 for dataset details. As shown in Table 5, our method drastically strengthens the capability of detecting background shifts; in contrast, it only slightly improves detecting semantic shifts, which indicates

Table 6: The OOD detection performance (FAR95) of different embedding approaches. The “ft” subscript denotes that the embedding model is fine-tuned on the in-distribution data for classification. The “+” subscript indicates better detection performance. The performance gain mainly comes from the task-agnostic general linguistic information in the holistic embeddings obtained by our pooling technique, in line with the probing analysis.

3.4 Comparison with Universal Sentence Embedding Approaches

Here we further show the advantage of our method Avg-Avg over two representative universal sentence embedding approaches, SentenceBERT (SBERT) (Reimers and Gurevych, 2019) and SimCSE (Gao et al., 2021) on OOD detection. For SBERT, we test the model trained on NLI (natural language inference) data (last-layer mean pooling is adopted as recommended in the original work); for SimCSE, we test the supervised model trained on NLI data (last-layer CLS pooling is adopted). The backbone model is RoBERTa-base in all methods. We also fine-tune the pre-trained models on the ID data and use the default pooling ways and our Avg-Avg to obtain embeddings from fine-tuned models for thorough comparison on OOD detection. As results in Table 6, when Avg-Avg is applied, it brings consistent improvements and beats both pre-trained and fine-tuned sentence embedding models using the default pooling way. These results corroborate the advantage of Avg-Avg as a specialized embedding method for OOD detection.

6679
and its effectiveness is confirmed by extensive experiments on different PLMs and benchmarks. Currently, its su-

This further supports our claim that Avg-Avg pooling strategies from the SST-2 test set (ID) and the RoBERTa-base model on SST-2 and visual-

Figure 2, in the representation space produced by Avg-Avg (Figure 2(b)) where is almost no overlap between ID and OOD instances, ID and OOD sam-

Figure 2: Visualization of the representations obtained for positive, negative instances in SST-2 and OOD ones (20 Newsgroups).

3.5 Embedding Visualization

To demonstrate the influence of the studied pooling strategies on the embedding space, we fine-tune the RoBERTa-base model on SST-2 and visualize instance embeddings corresponding to different pooling strategies from the SST-2 test set (ID) and an OOD test set (20 Newsgroups) using t-SNE (Van der Maaten and Hinton, 2008). As plotted in Figure 2, in the representation space produced by Avg-Avg (Figure 2(b)) where is almost no overlap between ID and OOD instances, ID and OOD samples are more sharply separated than in the space of default last-layer CLS embeddings (Figure 2(a)). This further supports our claim that Avg-Avg is better suited for OOD detection.

4 Related Works

4.1 Textual OOD Detection

OOD detection aims to detect abnormalities that come from a different distribution from the train-

4.2 Unsupervised Sentence Embedding

Unsupervised sentence embedding is a well-established area (Kiros et al., 2015; Pagliardini et al., 2017; Li et al., 2020; Reimers and Gurevych, 2019; Gao et al., 2021). Relevant to our work, Su et al. (2021) and Huang et al. (2021b) proposed to obtain better sentence embeddings via averaging token representations, layer combination, and a whitening operation. It is noteworthy that these embedding approaches are mainly studied for sentence matching and retrieval tasks. As far as we know, we are the first to study novel embedding ways to replace the default last-layer CLS pooling for boosting textual OOD detection.

5 Conclusion

In this work, we focus on how to derive sentence embeddings suitable for OOD detection from fine-tuned PLMs. Specifically, we introduce token averaging and layer combination to derive more holistic representations and substantially improve the capability of PLMs to detect OOD inputs. Moreover, our analysis shows that our approach helps preserve general linguistic information and benefits detecting background shifts. Overall, our work points out a new perspective that textual OOD detection can be enhanced by obtaining high-quality sentence embeddings, and we hope to extend this idea to training-time methods in future work.

Limitations

The contemporary solution Avg-Avg is primarily motivated by empirical observations and its effectiveness is confirmed by extensive experiments on different PLMs and benchmarks. Currently, its su-
priority lacks strict theoretical justifications and there is still a small performance gap between our method and the ideal upper bound as shown in Figure 1. In future work, we plan to explore theory-guaranteed embedding approaches to further boost the OOD detection ability of PLMs.

Ethical Considerations

Our work presents an efficient embedding method to enhance the OOD detection ability of NLP models. We believe that our proposal will help reduce security risks resulting from OOD inputs to NLP models deployed in the open-world environment. In addition, all experiments in this work are conducted on open datasets and our code is publicly available. While we do not expect any direct negative consequences to the work, we hope to continue exploring more efficient and robust sentence embedding approaches for textual OOD detection in future work.

Acknowledgement

We sincerely thank all the anonymous reviewers for their constructive comments and suggestions. This work is supported by Natural Science Foundation of China (NSFC) No. 62176002. Xu Sun is the corresponding author of this paper.

References


ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Seattle, Washington, USA, August 22-25, 2004, pages 168–177. ACM.


ence on Artificial Intelligence, volume 35, pages 13675–13682.


Table 7: Statistics of in-distribution text datasets. L denotes the average length of samples.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>#Classes</th>
<th>#Train</th>
<th>#Dev</th>
<th>#Test</th>
<th>L</th>
</tr>
</thead>
<tbody>
<tr>
<td>SST-2</td>
<td>2</td>
<td>6920</td>
<td>872</td>
<td>1821</td>
<td>19</td>
</tr>
<tr>
<td>IMDB</td>
<td>2</td>
<td>23000</td>
<td>2000</td>
<td>25000</td>
<td>230</td>
</tr>
<tr>
<td>TREC-10</td>
<td>6</td>
<td>4907</td>
<td>545</td>
<td>500</td>
<td>10</td>
</tr>
<tr>
<td>20 Newsgroups</td>
<td>20</td>
<td>10182</td>
<td>1132</td>
<td>7532</td>
<td>289</td>
</tr>
</tbody>
</table>

Table 8: Statistics of out-of-distribution text datasets. L denotes the average length of samples.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>#Test</th>
<th>L</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multi30k</td>
<td>1014</td>
<td>13</td>
</tr>
<tr>
<td>WMT16</td>
<td>2000</td>
<td>22</td>
</tr>
<tr>
<td>RTE</td>
<td>3000</td>
<td>48</td>
</tr>
<tr>
<td>SNLI</td>
<td>2000</td>
<td>21</td>
</tr>
</tbody>
</table>


A Dataset Statistics and Introduction

A.1 Datasets Used in Main Experiments

The statistics of in-distribution (ID) and out-of-distribution (OOD) textual datasets in main experiments (Section 3.1 and 3.2), including the number of classes, the dataset size, and the average length of samples, are given in Table 7 and 8, respectively. Here is a brief introduction to these datasets: Multi30k (Elliott et al., 2016) and WMT16 (Bojar et al., 2016) are parts of the English side data of English-German machine translation datasets; RTE (Dagan et al., 2005) and SNLI (Bowman et al., 2015) are the concatenations of the precise and respective hypotheses from NLI datasets.

A.2 Datasets Used In the Distribution Shift Analysis

Arora et al. (2021) categorized the distribution shifts in natural language data into two main types: background shifts and semantic shifts. We follow their division and study OOD detection performance under the setting where either kind of shift dominates in Section 3.3. The statistics of extra datasets used in the distribution shift analysis are given in Table 9. Here is a brief introduction to these datasets.

Background Shift Setting. Background shifts refer to the shift of background features (e.g., formality) that do not depend on the label. We consider domain shifts in sentiment classification datasets.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>#Test</th>
<th>L</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multi30k</td>
<td>1014</td>
<td>13</td>
</tr>
<tr>
<td>WMT16</td>
<td>2000</td>
<td>22</td>
</tr>
<tr>
<td>RTE</td>
<td>3000</td>
<td>48</td>
</tr>
<tr>
<td>SNLI</td>
<td>2000</td>
<td>21</td>
</tr>
</tbody>
</table>

A.2 Datasets Used In the Distribution Shift Analysis

The statistics of in-distribution (ID) and out-of-distribution (OOD) textual datasets in main experiments (Section 3.1 and 3.2), including the number of classes, the dataset size, and the average length of samples, are given in Table 7 and 8, respectively. Here is a brief introduction to these datasets: Multi30k (Elliott et al., 2016) and WMT16 (Bojar et al., 2016) are parts of the English side data of English-German machine translation datasets; RTE (Dagan et al., 2005) and SNLI (Bowman et al., 2015) are the concatenations of the precise and respective hypotheses from NLI datasets.
B.2 Contrastive Auxiliary Targets

Table 9: Statistics of extra datasets introduced for the distribution shift analysis. L denotes the average length of each sample.

<table>
<thead>
<tr>
<th>Dataset</th>
<th># Classes</th>
<th># Train</th>
<th># Dev</th>
<th># Test</th>
<th>L</th>
</tr>
</thead>
<tbody>
<tr>
<td>Customer Reviews (OOD)</td>
<td>2</td>
<td>-</td>
<td>-</td>
<td>1000</td>
<td>20</td>
</tr>
<tr>
<td>News Top-5 (ID)</td>
<td>5</td>
<td>68859</td>
<td>8617</td>
<td>8684</td>
<td>30</td>
</tr>
<tr>
<td>News Rest (OOD)</td>
<td>36</td>
<td>-</td>
<td>-</td>
<td>11402</td>
<td>29</td>
</tr>
<tr>
<td>CLINC (ID)</td>
<td>150</td>
<td>15000</td>
<td>3000</td>
<td>4500</td>
<td>8</td>
</tr>
<tr>
<td>CLINCood (OOD)</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>1000</td>
<td>9</td>
</tr>
</tbody>
</table>

Semantic Shift Setting. In this setting, OOD data are from the same task as ID data and share similar background characteristics, but belong to classes unseen during training. We use the News Category (Misra, 2018) and the CLINC (Larson et al., 2019b) datasets to create two ID/OOD pairs under the setting. Following Arora et al. (2021), we use the data from the five most frequent classes of the News Category as ID (News Top-5) and the data from the remaining 36 classes as OOD (News Rest). In the CLINC dataset for intent classification, there is a 150-class ID subset and an OOD test set CLINCood composed of utterances belonging to actions not supported by existing ID intents.

A.3 Probing Benchmarks

To probe the linguistic information contained in sentence embeddings, we use the probing tasks proposed by Conneau et al. (2018), which are grouped into three categories. For surface information, we use SentLen (sentence length) and WC (the presence of words); for syntactic information, we use BShift (sensitivity to word order), TreeDepth (the depth of the syntactic tree), and TopConst (the sequence of top-level constituents); for semantic information, we use Tense (tense), SubjNum and ObjNum (the subject/direct object number in the main clause), SOMO (the sensitivity to random replacement of a noun/verb), and CoordInv (the random swapping of coordinated clausal conjuncts). Each probing dataset contains 100k training samples, 10k validation samples, and 10k test samples. We use the SentEval toolkit (Conneau and Kiela, 2018) along with the recommended hyperparameter space to search for the best probing classifier according to the validation accuracy and report test accuracies.

B Details of Pretrained Language Model Fine-tuning

B.1 Vanilla Fine-tuning

We use the RoBERTa-base pretrained model (Liu et al., 2019) as the backbone to build text classifiers by fine-tuning it on the ID training data. We use a batch size of 16 and fine-tune the model for 5 epochs. The model is optimized with the Adam (Kingma and Ba, 2015) optimizer using a learning rate of 2e-5. We evaluate the model on the ID development set after every epoch and choose the best checkpoint as the final model. The setting is the same for other pretrained Transformers studied in the paper (RoBERTa-large, BERT-base-uncased, DistilRoBERTa-base, and ALBERT-base). DistilRoBERTa (Sanh et al., 2019) is a light distilled RoBERTa and ALBERT (Lan et al., 2019) is a lite BERT with factorized embedding parameterization and cross-layer parameter sharing.

B.2 Contrastive Auxiliary Targets

Zhou et al. (2021) introduced two alternatives of contrastive loss to boost textual OOD detection, i.e., the supervised contrastive loss and the margin-based contrastive loss. For a classification task with $C$ classes, given a batch of training examples $\{(x_i, y_i)\}_{i=1}^{M}$, where $x_i$ is the input and $y_i$ is the label, the supervised contrastive loss term $L_{\text{sc}}$ and the final optimization target $L$ can be formulated as:

$$L_{\text{sc}} = \sum_{i=1}^{M} \frac{-1}{M|P(i)|} \sum_{p \in P(i)} \log \frac{e^{\frac{1}{\tau} z_{i}^{T} z_{p}}}{\sum_{e \in A(i)} e^{\frac{1}{\tau} z_{i}^{T} z_{e}}}$$

$$L = L_{\text{ce}} + L_{\text{sc}}$$

where $A(i) = \{1, \ldots, M\} \setminus \{i\}$ is the set of all anchor instances, $P(i) = \{p \in A(i) : y_i = y_p\}$ is the set of anchor instances from the same class as $i$, $\tau$ is a temperature hyper-parameter, $z$ is the L2-normalized CLS embedding before the softmax layer, $L_{\text{ce}}$ is the cross entropy loss, and $\lambda$ is a positive coefficient. Following Zhou et al. (2021), we use $\tau = 0.3$ and $\lambda = 2$.

The margin-based loss term $L_{\text{margin}}$ and the final optimization target $L$ can be formulated as:
\[ L_{pos} = \sum_{i=1}^{M} \frac{1}{|P(i)|} \sum_{p \in P(i)} \| h_i - h_p \|^2, \]
\[ L_{neg} = \sum_{i=1}^{M} \frac{1}{|N(i)|} \sum_{n \in N(i)} (\xi - \| h_i - h_n \|^2)^+, \]
\[ L_{margin} = \frac{1}{dM} (L_{pos} + L_{neg}), \]
\[ \xi = \max_{i=1} \max_{p \in P(i)} \| h_i - h_p \|^2, \]
\[ L = L_{ce} + \lambda L_{margin}. \]

Here \( N(i) = \{ n \in A(i) : y_i \neq y_n \} \) is the set of anchor instances from other classes than \( y_i, h \in \mathbb{R}^d \) is the unnormalized CLS embedding before the softmax layer, \( \xi \) is the margin, \( d \) is the number of dimensions of \( h \), and \( \lambda \) is a positive coefficient. We use \( \lambda = 2 \) following Zhou et al. (2021).

Except for the loss term, we use the same hyper-parameters for these two tuning methods as vanilla tuning. Table 10 gives test accuracies on four ID datasets for the RoBERTa models tuned with vanilla cross-entropy loss (\( L_{ce} \)), supervised contrastive loss (\( L_{ce} + L_{uc} \)), and margin-based contrastive loss (\( L_{ce} + L_{margin} \)), where are not significant differences.

### B.3 Hardware Requirements

All the experiments (fine-tuning and inference) in this paper are conducted on a single NVIDIA TITAN RTX GPU, except that the fine-tuning of the RoBERTa-large model needs 4 TITAN RTX GPUs.

### C Definition of Evaluation Metrics for OOD Detection

For an input instance \( x \), the output of an OOD detector is the confidence score \( S(x) \) (a higher confidence score). A higher confidence score indicates that the detector tends to regard \( x \) as a normal ID sample. In real applications, system users need to choose a threshold \( \gamma \) and treat the OOD detection module as a binary classifier:

\[ G(x) = \begin{cases} 
\text{in}, & \text{if } S(x) \geq \gamma \\
\text{out}, & \text{if } S(x) < \gamma 
\end{cases} \]  (3)

Following previous works (Hendrycks and Gimpel, 2017; Zhou et al., 2021), we use the following two threshold-free metrics for evaluation:

**AUROC** is short for the area under the receiver operating curve. It plots the true positive rate (TPR) against the false positive rate (FPR) and can be interpreted as the probability that the model ranks a random positive (ID) example more highly than a random negative (OOD) example. A higher AUROC indicates better OOD detection performance.

**FAR95** is the probability for a negative example (OOD) to be mistakenly classified as positive (ID) when the TPR is 95%. A lower value indicates better detection performance.

### D OOD Detection Baselines

#### D.1 MSP

MSP (Hendrycks and Gimpel, 2017) is a classical baseline using the maximum softmax probability in the prediction outputs of the classifier as the confidence score, i.e., \( S(x) = \max_{y \in \mathcal{Y}} p_y(x) \).

#### D.2 Energy Score

Liu et al. (2020) proposed using free energy as a scoring function for OOD detection. For a classification problem with \( C \) classes, a multi-class classifier \( f(x) : \mathcal{X} \rightarrow \mathbb{R}^C \) can be interpreted from an energy-based perspective by viewing the logit output \( f_{yi}(x) \) corresponding to class \( yi \) as an energy function \( E(x, yi) = -f_{yi}(x) \). The free energy function \( E(x) \) for an input \( x \) is \( E(x) = \sum_{i=1}^{C} e^{f_{yi}(x)} \), and \( S(x) = -E(x) \).

#### D.3 LOF

Lin and Xu (2019) proposed identifying unknown user intents by feeding feature vectors to the density-based novelty detection algorithm, local outlier factor (LOF) (Breunig et al., 2000). We use the last-layer CLS vectors produced by the fine-tuned RoBERTa models as the input and train a LOF model following the implementation details of Lin and Xu (2019) on the ID training set and use the local density output as \( S(x) \).
AUROC↑ / FAR95↓ SST-2 IMDB TREC-10 20NG Avg
Maha Baseline 91.88 / 48.82 99.19 / 2.86 99.12 / 2.25 96.75 / 15.75 96.74 / 17.42
SE, token=CLS 94.68 / 26.09 99.94 / 0.05 99.75 / 0.19 99.47 / 2.62 98.45 / 7.24
SE, token=A VG 97.19 / 13.58 99.84 / 0.33 99.50 / 0.38 98.92 / 0.83 99.09 / 3.78
Avg-Avg (Ours) 97.75 / 10.67 99.87 / 0.21 99.70 / 1.35 99.70 / 1.35 99.25 / 3.12

Table 11: Comparison between score ensemble (SE) and Avg-Avg. The setting of backbone models and ID/OOD benchmarks is the same as that in Table 2.

D.4 Mahalanobis Distance

Mahalanobis distance score (Lee et al., 2018) is a representative distance-based OOD detection algorithm, which uses the sample distance to the nearest ID class in the embedding space as the OOD uncertainty measure. For a given feature extractor \( \psi \), the Mahalanobis distance score is defined as:

\[
S(x) = \max_{c \in \mathcal{T}} - (\psi(x) - \mu_c)^T \Sigma_c^{-1} (\psi(x) - \mu_c),
\]

where \( \psi(x) \) is the embedding vector of the input \( x \), \( \mu_c \) is the class centroid for a class \( c \), and \( \Sigma_c \) is the covariance matrix. The estimation of \( \mu_c \) and \( \Sigma_c \) is defined as follows:

\[
\mu_c = \frac{1}{N_c} \sum_{x \in D^{c}_{in}} \psi(x),
\]

\[
\Sigma = \frac{1}{N} \sum_{c \in \mathcal{T}} \sum_{x \in D^{c}_{in}} ((\psi(x) - \mu_c)(\psi(x) - \mu_c)^T),
\]

where \( D^{c}_{in} = \{ x \mid (x, y) \in D_{in}, y = c \} \) denotes the training samples belonging to the class \( c \), \( N \) is the size of the training set, and \( N_c \) is the number of training instances belonging to class \( c \).

E Comparison with Score Ensemble

Apart from the layer combination technique studied in the paper, there is another way to utilize intermediate representations for OOD detection: estimating the sample distance in the embedding space of each intermediate layer and taking their sum as the final OOD score. For the Mahalanobis distance score, the final ensemble score \( S(x) \) is defined as:

\[
S'(x) = \max_{c} - (\psi'(x) - \mu_c')^T \Sigma_c^{-1} (\psi'(x) - \mu_c'),
\]

\[
S(x) = \sum_{\ell} \alpha_{\ell} S'(x),
\]

where \( \psi'(x) \) denotes the output features at the \( \ell \)-th layer of neural networks, and \( \mu'_{\ell} \) and \( \Sigma_{\ell} \) are the class mean and the covariance matrix, correspondingly. The layer-wise weighting hyperparameter is \( \alpha_{\ell} \). In the original work (Lee et al., 2018), \( \alpha_{\ell} \) is tuned on a small validation set containing both ID and OOD for each OOD dataset, which is impractical in the setting of unsupervised OOD detection followed by recent works (OOD data is not available). Following Hsu et al. (2020), we use uniform weighting, i.e., \( S(x) = \sum_{\ell} S'(x) \), in the baselines for comparison.

We compare the performance of SE (score ensemble) and Avg-Avg and show the results in Table 11. We observe that SE also brings consistent improvements over the baseline using only last-layer CLS vectors. Without token averaging, SE slightly surpasses Avg-Avg on IMDB and TREC-10, but underperforms Avg-Avg significantly on SST-2 and 20NG; when token averaging is performed, SE only beats Avg-Avg on 20NG but underperforms on other three benchmarks, especially remarkably on SST-2. In view of the average performance on the four benchmarks, we can get Avg-Avg > SE (AVG) > SE (CLS). Considering that the class mean \( \mu'_{\ell} \) and the inverse of covariance matrix \( \Sigma_{\ell}^{-1} \) need to be estimated and stored for each layer in SE, Avg-Avg is also more convenient for deployment. So compared with SE, Avg-Avg enjoys both simplicity and advantages in performance.