kFolden: k-Fold Ensemble for Out-Of-Distribution Detection

Xiaoya Li[♠], Jiwei Li[♣♠], Xiaofei Sun[♠], Chun Fan[★▲▼]
Tianwei Zhang[♦], Fei Wu[♣], Yuxian Meng[♠], Jun Zhang[♥]
[♠]Shannon.AI
[▼]National Biomedical Imaging Center, Peking University
[★]Computer Center of Peking University, [▲]Peng Cheng Laboratory
[♦]Nanyang Technological University [♣]Zhejiang University, [♥]Tsinghua University
{xiaoya_li, jiwei_li, xiaofei_sun, yuxian_meng}@shannonai.com
fanchun@pku.edu.cn, tianwei.zhang@ntu.edu.sg
wufei@zju.edu.cn, jun-zhan19@mails.tsinghua.edu.cn

Abstract

Out-of-Distribution (OOD) detection is an important problem in natural language processing (NLP). In this work, we propose a simple yet effective framework kFolden, which mimics the behaviors of OOD detection during training without the use of any external data. For a task with k training labels, kFolden induces k sub-models, each of which is trained on a subset with $k-1$ categories with the left category masked unknown to the sub-model. Exposing an unknown label to the sub-model during training, the model is encouraged to learn to equally attribute the probability to the seen $k-1$ labels for the unknown label, enabling this framework to simultaneously resolve in- and out-distribution examples in a natural way via OOD simulations. Taking text classification as an archetype, we develop benchmarks for OOD detection using existing text classification datasets. By conducting comprehensive comparisons and analyses on the developed benchmarks, we demonstrate the superiority of kFolden against current methods in terms of improving OOD detection performances while maintaining improved in-domain classification accuracy. [1]

1 Introduction

Recent progress in deep neural networks has drastically improved accuracy in numerous NLP tasks (Sun et al., 2019; Raffel et al., 2019; Chai et al., 2020; He et al., 2020), but detecting out-of-distribution (OOD) examples from the in-domain (ID) examples is still a challenge for existing state-of-the-art deep NLP models. The ability of identifying OOD examples is critical for building reliable and trustworthy NLP systems for, say, text classification (Hendrycks and Gimpel, 2016; Mukherjee and Awadallah, 2020), question answering (Kamath et al., 2020) and neural machine translation (Kumar and Sarawagi, 2019). Existing works studying OOD detection in NLP often rely on external data (Hendrycks et al., 2018) to diversify model predictions and achieve better generality in OOD detection. The reliance on external data not only brings additional burden for data collection, but also results in the annoying issue in deciding which subset of external data to use: there is massive amount of external data and the using different subsets leads to different final results. Therefore, developing OOD detection system without external data is important towards building reliable NLP systems.

In this work, we propose a novel, simple yet effective framework, kFolden, short for a ***k*-Fold** **en**semble, to address OOD detection for NLP without the use of any external data. We accomplish this goal by simulating the process of detecting OOD examples *during training*. Concretely, for a standard NLP task with k labels for both training and test, we first obtain k separate sub-models, each of which is trained on a set of different $k-1$ labels with the left one being masked unknown to the model. We train each sub-model by jointly optimizing the cross entropy loss for the visible $k-1$ labels and the KL divergence loss between the predicted distribution and the uniform distribution for the left-one-out label. During test, we simply average the probability distributions produced by these k sub-modules and treat the result as the final probability estimate for a given input. Intuitively, if the input is an ID example, the final probability distribution will lay much of the weight on one of the k

[1] Corresponding author: Jun Zhang.

Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 3102–3115
November 7–11, 2021. ©2021 Association for Computational Linguistics

seen labels, but if the input is an OOD example, we expect the final probability distribution to get close to the uniform distribution, since each sub-model has tried to even its probability distribution when encountering unseen labels during training.

This training paradigm does not rely on any external data, and by mimicking the behaviors of distinguishing unseen labels from the seen, i.e., simulating the process of OOD detection during training, which is completed via the KL divergence loss, the framework naturally detects OOD examples and is able to perform reasonably better than other widely used strong OOD detection methods. Moreover, kFolden is complementary to existing post-hoc OOD detection methods, and combining both leads to the most performance boosts.

To facilitate OOD detection researches in NLP, we also construct benchmarks on top of four widely used text classification datasets: 20NewsGroups, Reuters, AG News and Yahoo!Answers. This created benchmark consists of 7 datasets with different levels of difficulty directed to two types of OOD examples: semantic shift and non-semantic shift, which differ in whether a shift is related to the inclusion of new semantic categories. The proposes benchmarks help comprehensively examine OOD detection methods, and we hope it can serve as a convenient and general tool for developing more robust and effective OOD detection models.

To summarize, the contributions of this work are:

- We propose a simple yet effective framework – kFolden, which simulates the process of OOD detection during training without using any external data.
- We construct benchmarks for OOD detection in text classification hoping for facilitating future related researches.
- We conduct comprehensive comparisons and analyses between existing methods and the proposed kFolden on the benchmark, and we show that kFolden achieves performance boosts regarding OOD detection while maintaining ID classification accuracy.

2 Related Work

Out-Of-Distribution Detection

Detecting OOD examples using deep neural models has gained substantial traction over recent years. Hendrycks and Gimpel (2016) proposed a baseline for misclassified and OOD examples by thresholding candidates based on the predicted softmax class probability. Lee et al. (2018) trained a classifier concurrent with a generator under the GAN framework (Goodfellow et al., 2014). The generator produces examples at the in-domain boundary and the classifier is forced to give lower confidence in predicting the classes for those examples. Hendrycks et al. (2018) leveraged real datasets instead of the generated examples, enabling the classifier to better generalize and detect anomalies. Liang et al. (2017) observed that temperature scaling and small perturbations lead to widened gaps between ID and OOD examples, for which they proposed proposed ODIN, a technique that makes OOD instances distinguishable by pulling apart the softmax scores of ID and OOD examples. Kamath et al. (2020) proposed to leverage the confidence estimate of a QA model to determine whether a question should be answered under domain shift to maintain a moderate accuracy. Hendrycks et al. (2019, 2020) showed that pretraining improves model robustness in terms of uncertainty estimation and OOD detection. Measuring model confidence has also exhibited power in detecting OOD examples (Lee et al., 2017a,b; DeVries and Taylor, 2018; Papadopoulos et al., 2021). This work differs from Hendrycks et al. (2020) mainly in that (1) they used a simple MaxProb-based method (Hendrycks and Gimpel, 2016) to estimate uncertainty while we propose a novel framework kFolden to improve OOD detection; and (2) they focused on comparing different NLP models on OOD generalization and shed light on the importance of pretraining for OOD robustness, whereas we highlight the merits of OOD simulation during training without the use of any external data, and construct a dedicated benchmark for text classification OOD detection.

Meta Learning in NLP

Meta learning (Thrun and Pratt, 2012; Andrychowicz et al., 2016; Nichol et al., 2018; Finn et al., 2017) tackles the problem of model learning in the domain with scarce data when large quantities of data are accessible in another related domain. Meta learning has been applied to considerable NLP tasks including semantic parsing (Huang et al., 2018; Guo et al., 2019; Sun et al., 2020), dialog generation (Song et al., 2019; Huang et al., 2020), text classification (Wu et al., 2019; Sun et al., 2020; Bansal et al., 2020; Lin et al., 2021) and machine translation (Gu et al., 2018). Our work is distantly related to meta learning in terms of the way we train kFolden by simulating the behaviors of predicting the unseen label during training, But we do not intend to achieve strong few-shot learning performances, which is the main goal of meta learning.

3 Task Definition

In this paper, we consider the problem of distinguishing betweem ID and OOD examples. We take text classification for illustration, and other tasks can be analogously resolved using the proposed kFolden framework. Let $\mathcal{D}^{\text{train}} = \{\boldsymbol{x}, y^{\text{train}}\}$ and $\mathcal{D}^{\text{test}} = \{\boldsymbol{x}, y^{\text{test}}\}$ denote the two sets respectively used for model training and test, where we assume the label space for training consists of k distinct labels $\mathcal{Y}^{\text{train}} = \{1, \cdots, k\}$ and all possible labels for test is the ones in $\mathcal{Y}^{\text{train}}$ plus t additional labels, i.e., $\mathcal{Y}^{\text{test}} = \{1, \cdots, k, k+1, \cdots, k+t\}$. Assume that a neural network f is trained on $\mathcal{D}^{\text{train}}$, and tested on $\mathcal{D}^{\text{test}}$.

We are interested in two situations when testing f on $\mathcal{D}^{\text{test}}$: (1) the current input example $\boldsymbol{x}$ has a gold label belonging to $\mathcal{Y}^{\text{train}}$ (i.e., $y^{\text{test}} \in \mathcal{Y}^{\text{train}}$), and (2) the input example's gold label does not belong to $\mathcal{Y}^{\text{train}}$ (i.e., $y^{\text{test}} \in \mathcal{Y}^{\text{test}} \backslash \mathcal{Y}^{\text{train}}$). For the former, we would like the model to achieve high accuracy because it has been trained on these ID examples; for the latter, we expect the model to figure out the current input is an OOD example. Hence, in this work, we mainly report the results from two aspects: accuracy on ID examples, and performances on OOD examples. The performance for OOD examples is evaluated via several targeted metrics, which will be introduced in experiments.

4 Method: k-Fold Ensemble

4.1 Training k Sub-Models as Simulation for OOD Detection

The core idea behind the proposed kFolden framework is to simulate the situation of encountering unseen labels at the training stage without the use of external data. To this end, we propose to train k independent sub-models $\{f_1, \cdots, f_k\}$, each of which is in order trained on a different subset of $k-1$ labels with the left label masked unknown to the model. Each sub-model is required to attain high accuracy on examples with the seen $k-1$ labels along with high uncertainty on examples with of masked label, and this is exactly what we would expect for OOD detection: we would like the model to accurately detect OOD examples while not harming performances on ID examples.

More specifically, assume we are training the i-th sub-model $f_i (1 \leq i \leq k)$, and thus the visible label set for training f_i would be $\mathcal{Y}^{\text{train}} \backslash \{i\}$. All training examples in $\mathcal{D}^{\text{train}}$ with label i now becomes unknown to f_i. For the visible $k-1$ labels, f_i should still achieve high accuracy as we want; but for the masked label i, f_i needs to give non-deterministic estimates when the input instance $\boldsymbol{x}$ has the ground-truth label i because the label i is masked and not found in the training set. This implies that the model can not determine which label $\boldsymbol{x}$ belongs to and may attribute it to an OOD example. These two considerations can be satisfied by jointly optimizing the following objective:

$$\mathcal{L} = \mathcal{L}_{\text{CE}} + \gamma \mathcal{L}_{\text{KL}} \tag{1}$$

where

$$\mathcal{L}_{\text{CE}} = \sum_{\substack{(\boldsymbol{x}, y^{\text{train}}) \in \mathcal{D}^{\text{train}} \\ y^{\text{train}} \in \mathcal{Y}^{\text{train}} \backslash \{i\}}} \texttt{CrossEntropy}(y^{\text{train}}, f_i(\boldsymbol{x})) \tag{2}$$

$$\mathcal{L}_{\text{KL}} = \sum_{\substack{(\boldsymbol{x}, y^{\text{train}}) \in \mathcal{D}^{\text{train}} \\ y^{\text{train}} = i}} \texttt{KL}(f_i(\boldsymbol{x}), \boldsymbol{u}) \tag{3}$$

γ is a hyper-parameter ranging over $[0, 1]$ and tuned on validation set. In the above equations, $\boldsymbol{u}$ is a uniform distribution. Eq.(2) is a standard cross entropy loss that requires the model to achieve accurate predictions on the visible labels, while Eq.(3) draws on the KL divergence to encourage the model to produce a probability distribution close to the uniform distribution $\boldsymbol{u}$ on the $k-1$ labels for the masked label. By jointly training on both loss functions, f_i will be able to detect the OOD label i while preserving non-reduced performances on other $k-1$ labels. We proceed with this process for all k sub-models, each with a different masked label. $f_i(\boldsymbol{x})$ takes as input $\boldsymbol{x}$ and outputs a probability distribution of dimensionality $k-1$. f_i can be implemented using any model backbone such as LSTM (Hochreiter and Schmidhuber, 1997), CNN (Kim, 2014), Transformer (Vaswani et al., 2017) and BERT (Devlin et al., 2018).

4.2 Sub-Model Ensemble

A single sub-model f_i will inevitably result in poor performances during test regarding the ID examples with label i. This is because for f_i, the masked label i during training will never have the chance to be predicted by the model, so that all the test examples with label i in $\mathcal{D}^{\text{test}}$ will be associated with possibly low probability, leading to overall reduced accuracy.

To tackle this issue, we adopt the idea of model ensemble: given an input $\boldsymbol{x}$, we first obtain k probability distributions $\{f_1(\boldsymbol{x}), \cdots, f_k(\boldsymbol{x})\}$ respectively produced by the k sub-models. In order to coordinate the label dimensions for different sub-models, we manually pad a zero dimension to each probability distribution at the corresponding masked position. For example, if $k = 4$ and the output from f_2 is $f_2(\boldsymbol{x}) = [f_2(\boldsymbol{x})_1, f_2(\boldsymbol{x})_2, f_2(\boldsymbol{x})_3, f_2(\boldsymbol{x})_4]$, then the padded output distribution would thus be $\tilde{f}_2(\boldsymbol{x}) = [f_2(\boldsymbol{x})_1, 0, f_2(\boldsymbol{x})_2, f_2(\boldsymbol{x})_3, f_2(\boldsymbol{x})_4]$. Next, we average all the k padded probability distributions, and take the result as the final probability estimate:

$$\tilde{f}(\boldsymbol{x}) = \frac{1}{k}\sum_{i=1}^{k}\tilde{f}_i(\boldsymbol{x}) \tag{4}$$

$\tilde{f}(\boldsymbol{x})$ is still a valid probability distribution and naturally remedies the shortcoming of a single sub-model: if $\boldsymbol{x}$ is an ID example, i.e., its ground-truth label y belongs to $\mathcal{Y}^{\text{train}}$, $\tilde{f}(\boldsymbol{x})$ will put most of the probability mass on one of the k labels; if $\boldsymbol{x}$ is an OOD example, $\tilde{f}(\boldsymbol{x})$ will get close to the uniform distribution because all sub-models comprising $\tilde{f}(\boldsymbol{x})$ will even their probability masses across all the k labels. After training, $\tilde{f}(\boldsymbol{x})$ can be used for ID evaluation and OOD evaluation simultaneously.

5 Benchmark Construction

Out-of-distribution data can be conceptually divided into two categories: non-semantic shift (NSS) and semantic shift (SS) and datasets (Hsu et al., 2020). They are different in terms of whether a shift is related to the inclusion of new semantic categories: the training and OOD test examples in the NSS dataset come from different sub-categories of the same broader category. For example, the training and OOD test sets in an NSS dataset are both from the "car" category, but examples in the training set are able "real car", e.g. "that's when they took out the fuel tank and poured it into a jug", and all OOD test are about "toy car", e.g. "Raleigh 2-year-old fills up toy car with 'gas' amidst shortage". For SS, the training and OOD test examples in the SS dataset come from completely different categories. For example, the training set contains labels "car" and "bicycle", and the test set has labels "train" and "plane", which have no intersections with training labels. In this paper, we construct both SS and NSS text classification benchmarks for OOD detection.

We construct benchmarks on multi-class topic classification datasets. The topic classification task has less vocabulary overlap between ID and OOD data. We use data from 20NewsGroups (Joachims, 1996), Reuters-21578[2], AG News (Del Corso et al., 2005) and Yahoo!Answers (Zhang et al., 2015). More details of the original datasets can be found at Appendix A. The statistics of the benchmark are present in Table 1.

We construct NSS benchmarks as follows:

20Newsgroups-6S This dataset is a modified version of 20Newsgroups. The original 20Newsgroups dataset has 20 newsgroups and each newsgroup (e.g., "comp.sys.ibm.pc.hardware") has a root subject topic (e.g., "comp"). We divide articles by its root subject and obtain 6 newsgroups ("comp", "rec", "sci", "religion", "politics" and "misc"). In this way, train and test data share the same root topic labeled but have different fine-grained topic labels. The training and ID test data are from 11 sub-classes in 20News, while OOD test data are from the rest 9 sub-classes.

AGNews-EXT This dataset is adapted from AG News and additional articles come from the AG Corpus. The original AG News dataset has 4 classes ("World", "Sports", "Business", "Sci/Tech"). The training and ID test data in AGNews-EXT come from the 4 class labels in AG News, and the OOD test data are from the AG Corpus but have the same class labels as in AG News.

Yahoo-AGNews-five This dataset contains a subset of Yahoo!Answers and a subset of AG Corpus. The original Yahoo!Answers dataset has 10 classes, and we use 5 of them ("Health", "Science & Mathematics", "Sports", "Entertainment & Music", "Business & Finance") for the training and ID test data. The OOD test data are selected from the 5 classes ("Health", "Sci/Tech", "Sports", "Entertainment", "Business") in AG Corpus.

We construct SS benchmarks as follows:

[2] http://kdd.ics.uci.edu/databases/reuters21578/reuters21578.html

	Non-Semantic Shift (NSS) Datasets			*Semantic Shift (SS) Datasets*			
	20News-6S	**AG-EXT**	**Yahoo-AG-five**	**Reuters-mK-nL**	**AG-FL**	**AG-FM**	**Yahoo-FM**
Adapted From	20News	AGNews&AGCorpus	Yahoo&AGCorpus	Reuters	AGNews&AGCorpus	AGNews&AGCorpus	Yahoo
# Labels in T	6	4	5	m	4	4	5
# Instances in T	8,283	112,400	675,000	$f(m, \text{train})$	116,000	116,000	680,000
# Labels in ID-V	6	4	5	m	4	4	5
# Instances in ID-V	1,034	7,600	25,000	$f(m, \text{valid})$	4,000	4,000	20,000
# Labels in OOD-V	6	4	5	n	4	4	5
# Instances in OOD-V	846	7,600	25,000	$f(n, \text{valid})$	4,000	4,000	20,000
# Labels in ID-T	6	4	5	m	4	4	5
# Instances in ID-T	1,034	7,600	25,000	$f(m, \text{test})$	4,000	4,000	25,000
# Labels in OOD-T	6	4	5	n	4	4	5
# Instances in OOD-T	846	7,600	25,000	$f(n, \text{test})$	4,000	4,000	25,000

Table 1: Statistics for the constructed benchmark. "T" is for "Training Set", "V" is for "Valid Set", and "T" is for "Test Set". All the data in each of the set are evenly distributed over the labels except 20News-6S. $f(m, \text{train/valid/test})$ means that the actual number is related to m and the corresponding train/valid/test set in the original Reuters-ModApte dataset.

Reuters-mK-nL This dataset is a modified version of Reuters. We first follow previous works (Yang and Liu, 1999; Joachims, 1998) to use the ModApte split[3] to remove documents belonging to multiple classes, and then considered only 10 classes ("Acquisitions", "Corn", "Crude", "Earn", "Grain", "Interest", "Money-fx", "Ship", "Trade" and "Wheat") with the highest numbers of training examples. The resulting dataset is called Reuters-ModApte. We train the model on a subset of Reuters-ModApte and test on the rest subset. Specifically, we train with m topic articles and test the model on the other $n = 10 - m$ topics. In this paper, we use five settings: $(m, n) = (9, 1)/(6, 4)/(5, 5)/(3, 7)/(2, 8)$.

AGNews-FL The dataset is adapted from AGNews and additional articles come from AG Corpus. In this setting, the training and ID test data are from the 4 classes ("World", "Sports", "Business", "Sci/Tech") in AGNews, and the OOD test data are from another 4 classes ("U.S.", "Europe", "Italia", "Software and Development") in AG Corpus.

AGNews-FM This dataset is adapted from AGNew and additional articles are taken from the AG Corpus. In this setting, the training and ID data are from the 4 classes ("World", "Sports", "Business", "Sci/Tech") in AGNews, and the OOD test data are from another 4 classes ("Entertainment", "Health", "Top Stories", "Music Feeds") in AG Corpus. This dataset is easier than AGNews-FL because the OOD labels are more distinct from the ID labels regarding the label semantics.

Yahoo!Answers-FM This dataset is modified from the Yahoo!Answers dataset. We use five topic articles ("Health", "Science & Mathematics", "Sports", "Entertainment & Music", "Business & Finance") for the training and ID tet data and use the other five unseen topics ("Society & Culture", "Education & Reference", "Computers & Internet", "Family & Relationships", "Politics & Government") for the OOD test data.

6 Experiments

6.1 Experimental Setups

We use both contextual and non-contextual model skeletons for experiments. We use CNN and BiLSTM as the non-contextual model backbones. We follow the CNN-non-static model (Kim, 2014) as the CNN implementation and the BiLSTM model is of a single layer. Both CNN and BiLSTM have $300d$ word vectors pretrained on Wikipedia 2014 using Glove (Pennington et al., 2014). The average of the hidden states of all words is used as the feature for classification. We trained the non-contextual models with a batch size of 32 and an initial learning rate of 0.001 using the Adam (Kingma and Ba, 2014). For contextual models, we use the officially pretrained BERT-uncased-base (Devlin et al., 2018) and RoBERTa-uncased-Base (Liu et al., 2019) for comparison. We use AdamW [4] to optimize all contextual models, with 0.01 weight decay and 1000 warmup steps. The learning rate was choosen in the range of $\{1e-5, 2e-5, 3e-5\}$. We use batch size in the range of $\{16, 24, 32\}$ for all experiments. And use dropout 0.2 for BERT and RoBERTa experiments.

[3] http://kdd.ics.uci.edu/databases/reuters21578/README.txt

[4] https://github.com/huggingface/transformers

	ID Metrics	OOD Metrics		
Model	**ACC↑**	**AUROC↑**	**AUPR↑**	**TNR@95TPR↑**
20Newsgroups-6S				
Vanilla				
CNN-init emb	77.76	50.22	58.91	29.27
BiLSTM-init emb	78.01	50.00	59.93	29.53
BERT	82.15	54.76	62.61	50.89
RoBERTa	83.40	57.41	66.79	59.15
RoBERTa+Mahalanobis	83.40	58.22	68.63	61.98
RoBERTa-Dropout	85.06	57.72	67.30	60.56
RoBERTa-Dropout+Mahalanobis	**85.06**	58.29	68.99	**62.40**
RoBERTa(6)	84.37	58.00	67.94	60.81
RoBERTa(6)+Mahalanobis	84.69	**58.42**	**69.05**	62.07
k**Folden**				
CNN-init emb	78.29	50.33	62.10	34.57
BiLSTM-init emb	78.30	50.48	60.86	34.94
BERT	84.12	56.77	64.85	53.46
RoBERTa	85.75	58.35	67.54	60.45
RoBERTa+Scaling	85.75	59.83	68.88	62.17
RoBERTa+Mahalanobis	**85.75**	**60.04**	**69.91**	**63.44**
AGNews-EXT				
Vanilla				
CNN-init emb	86.13	48.29	61.54	35.62
BiLSTM-init emb	87.38	48.56	62.15	35.88
BERT	92.24	51.35	63.68	49.63
RoBERTa	94.54	52.75	64.01	51.45
RoBERTa+Mahalanobis	94.54	55.37	65.94	54.60
RoBERTa-Dropout	95.13	52.74	64.32	52.47
RoBERTa-Dropout+Mahalanobis	95.13	55.67	66.32	55.10
RoBERTa(4)	95.22	53.91	65.68	53.08
RoBERTa(4)+Mahalanobis	**95.22**	**55.74**	**66.58**	**55.21**
k**Folden**				
CNN-init emb	88.30	49.31	62.18	37.20
BiLSTM-init emb	88.92	49.45	63.08	37.54
BERT	93.43	51.25	64.19	53.16
RoBERTa	95.62	53.87	65.76	54.98
RoBERTa+Scaling	95.62	55.19	66.28	55.09
RoBERTa+Mahalanobis	**95.62**	**56.07**	**67.81**	**55.48**
Yahoo-AGNews-five				
Vanilla				
CNN-init emb	77.45	79.26	58.50	43.94
BiLSTM-init emb	77.68	79.98	58.76	44.07
BERT	81.93	82.35	62.17	50.82
RoBERTa	82.54	84.98	63.46	50.94
RoBERTa+Mahalanobis	82.54	85.88	63.92	51.96
RoBERTa-Dropout	84.04	84.29	63.36	51.03
RoBERTa-Dropout+Mahalanobis	84.04	85.95	64.17	51.39
RoBERTa(5)	84.10	85.01	64.14	51.22
RoBERTa(5)+Mahalanobis	**84.10**	**86.23**	**64.37**	**53.11**
k**Folden**				
CNN-init emb	79.23	81.12	61.09	45.82
BiLSTM-init emb	78.04	82.33	62.88	45.90
BERT	83.23	84.09	63.11	52.95
RoBERTa	84.45	85.61	64.15	52.22
RoBERTa+Scaling	84.45	86.69	64.87	54.39
RoBERTa+Mahalanobis	**84.45**	**86.92**	**64.92**	**56.24**

Table 2: Results of Non-Semantic Shift (NSS) datasets. The number in the bracket (k) denotes averaging k model predictions and k equals to the number of labels in the training dataset.

6.2 Baselines

We choose the following OOD detection methods for comparison:

MSP: The Maximum Softmax Probability method proposed by Hendrycks and Gimpel (2016). It uses the maximum probability in the final probability distribution over labels as the prediction score. If the maximum probability is under some specified threshold $\varphi \in [0, 1]$, then the example would be classified as OOD. We tune the threshold on the dev set. This is the default setting for all model backbones.

	ID Metrics	OOD Metrics		
Model	**ACC↑**	**AUROC↑**	**AUPR↑**	**TNR@95TPR↑**
Reuters-7K-3L				
Vanilla				
CNN-init emb	62.04	64.76	53.49	49.23
BiLSTM-init emb	60.89	66.41	55.55	48.58
BERT	63.25	66.83	60.28	50.66
RoBERTa	65.88	67.37	63.30	51.95
RoBERTa+Mahalanobis	65.88	68.34	64.33	52.73
RoBERTa-Dropout	66.16	69.04	64.18	52.09
RoBERTa-Dropout+Mahalanobis	66.16	69.86	64.25	52.90
RoBERTa(7)	66.31	69.31	64.57	52.81
RoBERTa(7)+Mahalanobis	**66.31**	**69.89**	**64.82**	**53.46**
k**Folden**				
CNN-init emb	62.94	65.08	54.28	50.27
BiLSTM-init emb	61.05	67.81	56.98	49.96
BERT	65.45	68.14	61.11	51.79
RoBERTa	66.72	69.70	64.74	53.62
RoBERTa+Scaling	66.72	70.03	65.39	53.98
RoBERTa+Mahalanobis	**66.72**	**70.52**	**65.81**	**54.91**
AGNews-FL				
Vanilla				
CNN-init emb	80.55	62.94	52.70	30.54
BiLSTM-init emb	81.36	63.71	54.77	31.90
BERT	85.58	64.55	54.49	42.84
RoBERTa	87.19	65.52	55.48	45.89
RoBERTa+Mahalanobis	87.19	66.20	56.45	46.95
RoBERTa-Dropout	87.27	65.61	56.38	46.06
RoBERTa-Dropout+Mahalanobis	87.27	66.53	57.11	46.89
RoBERTa(4)	87.55	65.81	56.89	46.19
RoBERTa(4)+Mahalanobis	**87.55**	**66.48**	**57.49**	**46.92**
k**Folden**				
CNN-init emb	82.21	63.45	53.98	34.71
BiLSTM-init emb	84.33	64.44	55.01	35.68
BERT	87.20	65.19	55.39	45.39
RoBERTa	88.03	66.29	57.39	46.27
RoBERTa+Scaling	88.03	66.84	58.07	46.75
RoBERTa+Mahalanobis	**88.03**	**66.89**	**58.26**	**47.16**
AGNews-FM				
Vanilla				
CNN-init emb	79.81	79.63	53.50	54.72
BiLSTM-init emb	82.51	79.46	52.86	55.33
BERT	83.40	80.63	56.79	59.84
RoBERTa	85.62	82.53	58.84	60.36
RoBERTa+Mahalanobis	85.62	83.04	59.96	62.26
RoBERTa-Dropout	87.59	82.64	59.76	60.86
RoBERTa-Dropout+Mahalanobis	87.59	83.14	59.23	61.88
RoBERTa(4)	88.16	82.85	60.44	61.95
RoBERTa(4)+Mahalanobis	**88.16**	**83.27**	**60.82**	**62.34**
k**Folden**				
CNN-init emb	80.77	79.83	55.63	55.69
BiLSTM-init emb	83.43	80.23	57.40	55.57
BERT	84.55	81.35	58.19	62.89
RoBERTa	88.92	83.61	60.88	63.42
RoBERTa+Scaling	88.92	84.04	61.27	63.73
RoBERTa+Mahalanobis	**88.92**	**84.31**	**61.48**	**64.29**
Yahoo!Answers-FM				
Vanilla				
CNN-init emb	89.44	80.36	69.49	55.01
BiLSTM-init emb	90.57	79.42	68.43	55.49
BERT	93.25	82.71	74.55	57.82
RoBERTa	94.73	83.81	76.47	58.62
RoBERTa+Mahalanobis	94.73	84.51	77.38	59.86
RoBERTa-Dropout	95.13	84.46	77.09	59.05
RoBERTa-Dropout+Mahalanobis	95.13	84.90	77.50	59.99
RoBERTa(5)	95.16	84.78	77.42	59.18
RoBERTa(5)+Mahalanobis	**95.16**	**85.06**	**77.92**	**60.28**
k**Folden**				
CNN-init emb	90.38	81.92	70.82	57.49
BiLSTM-init emb	91.42	82.84	72.81	58.06
BERT	94.74	84.15	76.92	58.34
RoBERTa	95.56	85.50	78.52	59.10
RoBERTa+Scaling	95.56	85.66	78.82	59.95
RoBERTa+Mahalanobis	**95.56**	**85.83**	**78.88**	**61.70**

Table 3: Results of Semantic Shift (SS) datasets. The number in the bracket (k) denotes averaging k model predictions and k equals to the number of labels in the training dataset.

Scaling: The temperature scaling (Guo et al., 2017) method leverages a temperature $T > 0$ to sharpen or widen the probability distribution,

Hyperparameter	Values to select
batch size	{16, 24, 32, 48}
dropout	{0.1, 0.2, 0.3}
weight decay	{0, 0.01}
max epochs	{3, 5, 8}
warmup ratio	{0, 0.1, 0.05}
learning rate	{1e-5, 2e-5, 3e-5}
learning rate decay	linear
gradient clip	1.0
MSP threshold φ	{0, 0.001, 0.01, 0.05, 0.1, 0.2}
Scaling temperature T	{1, 10, 100, 1000, 5000 }
Scaling threshold φ	{0, 0.0005, 0.001, 0.0015, 0.002, 0.005, 0.01, 0.05, 0.1, 0.2}
number of passes in **Dropout**	{5, 10, 15, 20, 30}

Table 4: The range of hyperparameter values.

and then treats the maximum probability as the final score. The temperature T is chosen from $\{1, 10, 100, 1000, 5000\}$ and is selected on the OOD validation set.

Mahalanobis: Lee et al. (2018) defined the confidence score using the Mahalanobis distance of a test example $\boldsymbol{x}$ with respect to the closest class-conditional distribution, which can be expressed as: $\text{score}(\boldsymbol{x}) = \min_c (\psi(\boldsymbol{x}) - \boldsymbol{\mu}_c)^\top \boldsymbol{\Sigma}^{-1} (\psi(\boldsymbol{x}) - \boldsymbol{\mu}_c)$, where $\psi(\boldsymbol{x})$ is the vector representation of the input $\boldsymbol{x}$, $\boldsymbol{\mu}_c = \frac{1}{N_c} \sum_{\boldsymbol{x} \in \mathcal{D}^c} \psi(\boldsymbol{x})$ is the centroid for class c in the valid set $\mathcal{D}^{\text{valid}}$ and $\boldsymbol{\Sigma} = \frac{1}{N} \sum_c \sum_{\boldsymbol{x} \in \mathcal{D}^c} (\psi(\boldsymbol{x}) - \boldsymbol{\mu}_c)(\psi(\boldsymbol{x}) - \boldsymbol{\mu}_c)^\top$ is the co-variance matrix. N_c is the number of instances belongs to class c in $\mathcal{D}^{\text{valid}}$.

Dropout: Gal and Ghahramani (2016) casted dropout training as Bayesian inference for neural networks and obtained multiple predictions by running the model multiple times with dropout opened for a fixed input. These predictions are then averaged, giving the final probability distribution. Note that we can combine this method with the above three approaches.

More details regarding hyperparameter selection are present in Table 4. Since the proposed strategy uses the ensemble of K models, we also implement an ensemble of k vanilla models.

6.3 Metrics

We use accuracy (ACC) to evaluate model performances on the in-distribution testset and follow previous works (Hendrycks and Gimpel, 2016; Hsu et al., 2020; Lee et al., 2018) to employ three metrics for the OOD detection task, including AUROC, AUPR$_{out}$, TNR@95TPR.

AUROC: The AUROC is short for area under the receiver operating characteristic curve. The ROC curve is a graph plotting true negative rate against the false positive rate = FP/(FP+TN) by varying a threshold. This score is a threshold-independent evaluation metric and can be interpreted as the probability that a positive example has a greater detector score/value than a negative example (Fawcett, 2006). A random classifier has an AUROC score of 50%. A higher AUROC value indicates a better OOD detection performance.

AUPR$_{out}$: The AUPR is short for the area under the precision-recall curve. The precison-recall curve is a graph plotting the precision=TP/(TP+FP) against recall=TP/(TP+FN) by varying a threshold. AUPR$_{out}$ requires taking out-of-distribution data as the positive class. It is more suitable for highly imbalanced data compared to AUROC.

TNR@95TPR: The TNR@95TPR is short for true negative rate (TNR) at 95% true positive rate (TPR). The TNR@95TPR measures the true negative rate (TNR = TN/(FP+TN)) when the true positive rate (TPR = TP/(TP+FN)) is 95%, where TP, TN, FP and FN denotes true positive, true negative, false positive and false negative, respectively. It can be interpreted as the probability that an example predicted incorrectly is misclassified as a corrected prediction when TPR is equal to 95%.

6.4 Results

Experimental results for non-semantic shift and semantic shift benchmarks are shown in Table 2 and Table 3, respectively. The first observation is that contextual models (BERT and RoBERTa) can achieve significantly better performances on both in-distribution and out-of-distribution datasets than non-contextual models (e.g., CNN, LSTM). The second observation is that existing methods including Scaling, Mahalanobis and Dropout can improve ID and OOD performances. The proposed kFolden framework introduces performance boost over the ensemble of its corresponding vanilla model (e.g., CNN, LSTM, Bert and RoBerta) in both ID and

OOD evaluations. Additionally, we also find that kFolden is a flexible and general framework, which can be combined to existing OOD detection methods such as Mahalanobis, scaling and dropout, and can introduce addition performance boosts in OOD detection.

It is interesting to see that the improvements on SS datasets are greater than on NSS datasets when augmenting with the kFolden framework. This is because compared to NSS tasks, SS poses more variability in data distributions and requires a better generality from ID to OOD samples. kFolden serves this purpose well since it performs in a way as OOD simulation during training, which naturally addresses ID classification and OOD detection at the same time during training. This training paradigm wins better results for kFolden on SS data.

6.5 The Ratio of Unseen Labels

In this subsection, we explore the effect of unseen categories at different ratios. We use RoBERTa as the model backbone and conduct experiments on Reuters-mK-nL datasets, including 9K-1L, 6K-4L, 5K-5L, 3K-7L and 2K-8L. We use accuracy and the error rate as evaluation metrics. The error rate represents the the proportion of OOD examples that are incorrectly classified to an in-distribution label, i.e., the maximum class probability is above the threshold tuned on the valid set. Experimental results are shown in Table 5. As we can see from Table 5, the overall trend is that the error rate increases as more unseen text categories are added to the out-of-distribution test set. Regarding specific models, we find that kFolden always outperforms Dropout, and the combination of kFolden and Mahalanobis leads to the best performance. We speculate that this is because unlike Dropout which relies on the masking patterns within the neural network, the kFolden framework straightforwardly performs at the output, or the training objective level using the training data. This gives a direct learning signal for the model to learn to distinguish OOD examples.

Model	AUPR↑	Error Rate↓
	Reuters-9K-1L	
RoBERTa	79.77	36.61
RoBERTa+Dropout	80.07	32.74
kFolden RoBERTa	81.53	30.63
kFolden RoBERTa+mahal	**81.68**	**29.75**
	Reuters-6K-4L	
RoBERTa	78.52	36.26
RoBERTa+Dropout	79.73	36.13
kFolden RoBERTa	80.83	35.76
kFolden RoBERTa+mahal	**82.74**	**35.49**
	Reuters-5K-5L	
RoBERTa	89.56	42.83
RoBERTa+Dropout	90.25	41.36
kFolden RoBERTa	91.76	40.99
kFolden RoBERTa+mahal	**92.08**	**40.76**
	Reuters-3K-7L	
RoBERTa	95.64	46.27
RoBERTa+Dropout	96.14	45.89
kFolden RoBERTa	96.75	44.82
kFolden RoBERTa+mahal	**96.83**	**43.69**
	Reuters-2K-8L	
RoBERTa	97.35	58.14
RoBERTa+Dropout	97.56	57.62
kFolden RoBERTa	97.83	56.80
kFolden RoBERTa+mahal	**97.91**	**56.06**

Table 5: Results on Reuters-mK-nL OOD test sets. The Reuters dataset contains 10 label categories. We use m to represent the number of labels in ID training set and n for the number of categories in OOD testset.

7 Conclusion

In this paper, we propose a simple yet effective framework kFolden for OOD detection. It works by mimicking the behaviors of detecting out-of-distribution examples during training without the use of any external data. We also develop a benchmark on top of existing widely used datasets for text classification OOD detection. This benchmark contains both semantic shift and non-semantic shift data, which would benefit a comprehensive examination to the ability of OOD detection methods. Through experiments and analyses, we show that the proposed kFolden framework outperforms strong OOD detection baselines on the constructed benchmark, and combining kFolden and other post-hoc methods leads to the most performance gains. We hope the proposed method and the created benchmark can facilitate further researches in related areas.

Acknowledgement

This work was supported by Key-Area Research and Development Program of Guangdong Province (No. 2019B121204008). We thank the High-Performance Computing Platform at Peking University and the PCNL Cloud Brain for providing

platforms of data analysis and model training.

References

Marcin Andrychowicz, Misha Denil, Sergio Gomez, Matthew W Hoffman, David Pfau, Tom Schaul, Brendan Shillingford, and Nando De Freitas. 2016. Learning to learn by gradient descent by gradient descent. *arXiv preprint arXiv:1606.04474*.

Trapit Bansal, Rishikesh Jha, Tsendsuren Munkhdalai, and Andrew McCallum. 2020. Self-supervised meta-learning for few-shot natural language classification tasks. *arXiv preprint arXiv:2009.08445*.

Duo Chai, Wei Wu, Qinghong Han, Fei Wu, and Jiwei Li. 2020. Description based text classification with reinforcement learning. In *International Conference on Machine Learning*, pages 1371–1382. PMLR.

Gianna M Del Corso, Antonio Gulli, and Francesco Romani. 2005. Ranking a stream of news. In *Proceedings of the 14th international conference on World Wide Web*, pages 97–106. ACM.

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

Terrance DeVries and Graham W Taylor. 2018. Learning confidence for out-of-distribution detection in neural networks. *arXiv preprint arXiv:1802.04865*.

Tom Fawcett. 2006. An introduction to ROC analysis. *Pattern Recognit. Lett.*, 27(8):861–874.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, pages 1126–1135. PMLR.

Yarin Gal and Zoubin Ghahramani. 2016. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059. PMLR.

Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial networks. *arXiv preprint arXiv:1406.2661*.

Jiatao Gu, Yong Wang, Yun Chen, Kyunghyun Cho, and Victor OK Li. 2018. Meta-learning for low-resource neural machine translation. *arXiv preprint arXiv:1808.08437*.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. 2017. On calibration of modern neural networks. In *International Conference on Machine Learning*, pages 1321–1330. PMLR.

Daya Guo, Duyu Tang, Nan Duan, Ming Zhou, and Jian Yin. 2019. Coupling retrieval and meta-learning for context-dependent semantic parsing. *arXiv preprint arXiv:1906.07108*.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*.

Dan Hendrycks and Kevin Gimpel. 2016. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *arXiv preprint arXiv:1610.02136*.

Dan Hendrycks, Kimin Lee, and Mantas Mazeika. 2019. Using pre-training can improve model robustness and uncertainty. In *International Conference on Machine Learning*, pages 2712–2721. PMLR.

Dan Hendrycks, Xiaoyuan Liu, Eric Wallace, Adam Dziedzic, Rishabh Krishnan, and Dawn Song. 2020. Pretrained transformers improve out-of-distribution robustness. *arXiv preprint arXiv:2004.06100*.

Dan Hendrycks, Mantas Mazeika, and Thomas Dietterich. 2018. Deep anomaly detection with outlier exposure. *arXiv preprint arXiv:1812.04606*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Yen-Chang Hsu, Yilin Shen, Hongxia Jin, and Zsolt Kira. 2020. Generalized odin: Detecting out-of-distribution image without learning from out-of-distribution data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10951–10960.

Po-Sen Huang, Chenglong Wang, Rishabh Singh, Wen-tau Yih, and Xiaodong He. 2018. Natural language to structured query generation via meta-learning. *arXiv preprint arXiv:1803.02400*.

Yi Huang, Junlan Feng, Min Hu, Xiaoting Wu, Xiaoyu Du, and Shuo Ma. 2020. Meta-reinforced multi-domain state generator for dialogue systems. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7109–7118.

Thorsten Joachims. 1996. A probabilistic analysis of the rocchio algorithm with tfidf for text categorization. Technical report, Carnegie-mellon univ pittsburgh pa dept of computer science.

Thorsten Joachims. 1998. Text categorization with support vector machines: Learning with many relevant features. In *European conference on machine learning*, pages 137–142. Springer.

Amita Kamath, Robin Jia, and Percy Liang. 2020. Selective question answering under domain shift. *arXiv preprint arXiv:2006.09462*.

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.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Aviral Kumar and Sunita Sarawagi. 2019. Calibration of encoder decoder models for neural machine translation. *arXiv preprint arXiv:1903.00802*.

Kimin Lee, Changho Hwang, KyoungSoo Park, and Jinwoo Shin. 2017a. Confident multiple choice learning. In *International Conference on Machine Learning*, pages 2014–2023. PMLR.

Kimin Lee, Honglak Lee, Kibok Lee, and Jinwoo Shin. 2017b. Training confidence-calibrated classifiers for detecting out-of-distribution samples. *arXiv preprint arXiv:1711.09325*.

Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. 2018. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. *arXiv preprint arXiv:1807.03888*.

Shiyu Liang, Yixuan Li, and Rayadurgam Srikant. 2017. Enhancing the reliability of out-of-distribution image detection in neural networks. *arXiv preprint arXiv:1706.02690*.

Yuxiao Lin, Yuxian Meng, Xiaofei Sun, Qinghong Han, Kun Kuang, Jiwei Li, and Fei Wu. 2021. Bertgcn: Transductive text classification by combining gcn and bert. *arXiv preprint arXiv:2105.05727*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Subhabrata Mukherjee and Ahmed Hassan Awadallah. 2020. Uncertainty-aware self-training for text classification with few labels. *arXiv preprint arXiv:2006.15315*.

Alex Nichol, Joshua Achiam, and John Schulman. 2018. On first-order meta-learning algorithms. *arXiv preprint arXiv:1803.02999*.

Aristotelis-Angelos Papadopoulos, Mohammad Reza Rajati, Nazim Shaikh, and Jiamian Wang. 2021. Outlier exposure with confidence control for out-of-distribution detection. *Neurocomputing*, 441:138–150.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1532–1543. ACL.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*.

Yiping Song, Zequn Liu, Wei Bi, Rui Yan, and Ming Zhang. 2019. Learning to customize model structures for few-shot dialogue generation tasks. *arXiv preprint arXiv:1910.14326*.

Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. 2019. How to fine-tune bert for text classification? In *China National Conference on Chinese Computational Linguistics*, pages 194–206. Springer.

Yibo Sun, Duyu Tang, Nan Duan, Yeyun Gong, Xiaocheng Feng, Bing Qin, and Daxin Jiang. 2020. Neural semantic parsing in low-resource settings with back-translation and meta-learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8960–8967.

Sebastian Thrun and Lorien Pratt. 2012. *Learning to learn*. Springer Science & Business Media.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Jiawei Wu, Wenhan Xiong, and William Yang Wang. 2019. Learning to learn and predict: A meta-learning approach for multi-label classification. *arXiv preprint arXiv:1909.04176*.

Yiming Yang and Xin Liu. 1999. A re-examination of text categorization methods. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 42–49.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. *arXiv preprint arXiv:1509.01626*.

A Dataset Details

A.1 Original Datasets

In this paper, We use data from 20NewsGroups (Joachims, 1996), Reuters-21578[5], AG News (Del Corso et al., 2005) and Yahoo!Answers (Zhang et al., 2015) to construct our evaluation benchmark. Details regarding these four datasets are present below:

- **20Newsgroups**[6]: 20Newsgroups is a collection of approximate 20,000 newsgroup documents, partitioned (nearly) evenly across 20 different newsgroups. Each newsgroup corresponds to a different topic. Some of the newsgroups are very closely related to each other (e.g., "comp.sys.pc.hardware" and

[5] http://kdd.ics.uci.edu/databases/reuters21578/reuters21578.html

[6] https://kdd.ics.uci.edu/databases/20newsgroups/20newsgroups.html

“comp.sys.mac.hardware”), while others are highly unrelated (e.g., “misc.forsale” and “soc.religion.christian”).

- **AG News**[7]: AG News is a subdataset of AG’s corpus of news articles constructed by assembling titles and description fields of articles from the four largest classes ("World", "Sports", "Business", "Sci/Tech") of AG Corpus. AG News contains 30,000 training and 1,900 test samples per class.
- **Yahoo!Answers**[8]: Yahoo!answers was constructed by Zhang et al. (2015) and composed of 10 largest main categories from Yahoo!Answers Comprehensive Questions and the Answers version 1.0 dataset. Each class contains 140,000 training samples and 5,000 testing samples. Labels in the dataset include "Society & Culture", "Science & Mathematics", "Health", "Education & Reference", "Computers & Internet", "Sports", "Business & Finance", "Entertainment & Music", "Family & Relationships", and "Politics & Government".
- **Reuters-21578**[9]: Reuters-21578 is a collection of 10,788 documents from the Reuters financial newswire service, partitioned into a training set with 7,769 documents and a test set with 3,019 documents. The distribution of categories in the Reuters-21578 corpus is highly skewed, with 36.7% of the documents in the most common category, and only 0.0185% (2 documents) in each of the five least common categories. There are 90 categories in the corpus. Each document belongs to one or more categories. The average number of categories per document is 1.235, and the average number of documents per category is about 148, or 1.37% of the corpus.

A.2 Benchmark Construction

We construct our NSS benchmarks as follows:

20Newsgroups-6S This dataset is a modified version of 20Newsgroups. The original 20Newsgroups dataset has 20 newsgroups and each newsgroup (e.g., "comp.sys.ibm.pc.hardware") has a root subject topic (e.g., "comp"). We divide articles by its root subject and obtain 6 newsgroups ("comp", "rec", "sci", "religion", "politics" and "misc"). For example, the original label "comp.sys.ibm.pc.hardware" becomes "comp". Hence, train and test data share the same labels but may come from different data distributions. Data in the five sets do not overlap. We show the used classes for each of the following sets in Table 6.

Label	Train&ID-X	OOD-X
comp	comp.graphics comp.sys.ibm.pc.hardware comp.os.ms-windows.misc	comp.sys.mac.hardware comp.windows.x
rec	rec.autos rec.motorcycles	rec.sport.baseball rec.sport.hockey
sci	sci.crypt sci.electronics	sci.med sci.space
religion	talk.religion.misc	alt.atheism soc.religion.christian
politics	talk.politics.guns talk.politics.misc	talk.politics.mideast
misc	misc.forsale	

Table 6: Merging labels from 20News for 20News-6S.

- **TrainingSet** We use 8,283 articles from the trainset in 20Newsgroups belonging to 11 sub-classes. Each class contains 753 articles.
- **ID-ValidSet** We use 1,034 articles from the trainset in 20Newsgroups belonging to 11 sub-classes. Each class contains 94 articles.
- **ID-TestSet** We use 1,034 articles from the testset in 20Newsgroups belonging to 11 sub-classes. Each class contains 94 articles.
- **OOD-ValidSet** We use 846 articles from the trainset in 20Newsgroups belonging to the other 9 sub-classes in 20Newsgroups. Each class contains 94 articles.
- **OOD-TestSet** We use 846 articles from the testset in 20Newsgroups belonging to the other 9 sub-classes in 20Newsgroups. Each class contains 94 articles.

AGNews-EXT This dataset contains data from AG-News and additional articles from AG Corpus. In this setting, the training and ID data are from the same 4 labels ("World", "Sports", "Business", "Sci/Tech"). OOD data are from the same 4 labels but use articles in AG Corpus intead of AG-News. Data in the five sets do not overlap.

- **TrainingSet** We use 112,400 articles from the trainset in AG-News with 4 classes. Each class contains 28,100 articles.

[7] http://groups.di.unipi.it/~gulli/AG_corpus_of_news_articles.html

[8] https://drive.google.com/drive/folders/0Bz8a_Dbh9Qhbfll6bVpmNUtUcFdjYmF2SEpmZUZUcVNiMUw1TWN6RDV3a0JHT3kxLVhVR2M

[9] http://kdd.ics.uci.edu/databases/reuters21578/reuters21578.html

- **ID-ValidSet** We use 7,600 articles from the trainset in AG-News with the same 4 classes as TrainingSet. Each class has 1,900 articles.
- **ID-TestSet** We use 7,600 articles from the testset in AG-News with the same 4 classes as TrainingSet. Each class has 1,900 articles.
- **OOD-ValidSet** We assemble titles and description fields of articles in AG Corpus from the same 4 classes as TrainingSet. Each class has 1,900 articles.
- **OOD-TestSet** We assemble titles and description fields of articles in AG Corpus from the same 4 classes as TrainingSet. Each class has 1,900 articles.

Yahoo-AGNews-five This dataset contains a subset of Yahoo!Answers and a subset of AG Corpus. The original Yahoo!Answers dataset has 10 classes, and we use 5 of them ("Health", "Science & Mathematics", "Sports", "Entertainment & Music", "Business & Finance") for the training and ID data. The OOD data are from the 5 classes ("Health", "Sci/Tech", "Sports", "Entertainment", "Business") in AG Corpus. Data in the five sets do not overlap.

- **TrainingSet** We use 675,000 articles from the trainset in Yahoo!Answers with 5 classes. Each class contains 135,000 articles.
- **ID-ValidSet** We use 25,000 articles from the trainset in Yahoo!Answers with the same 5 classes as TrainingSet. Each class contains 5,000 articles.
- **ID-TestSet** We use 25,000 articles from the testset in Yahoo!Answers with the same 5 classes as TrainingSet. Each class contains 5,000 articles.
- **OOD-ValidSet** We assemble titles and description fields of articles in AG Corpus from the same 5 classes as TrainingSet. Each class contains 5,000 articles.
- **OOD-TestSet** We assemble titles and description fields of articles in AG Corpus from the same 5 classes as TrainingSet. Each class contains 5,000 articles.

We construct SS benchmarks as follows:

Reuters-mK-nL This dataset is a modified version of Reuters. We first follow previous works (Yang and Liu, 1999; Joachims, 1998) to use the ModApte split[10] to remove documents belonging to multiple classes, and then considered only 10 classes ("Acquisitions", "Corn", "Crude", "Earn", "Grain", "Interest", "Money-fx", "Ship", "Trade" and "Wheat") with the highest numbers of training examples. The resulting dataset is called Reuters-ModApte. We train the model on a subset of Reuters-ModApte and test on the rest subset. Specifically, we train with m topic articles and test the model on the other n ($n = 10 - m$) topic articles. In this paper, we use five settings: Reuters-9K-1L, Reuters-6K-4L, Reuters-5K-5L, Reuters-3K-7L and Reuters-2K-8L. This task is difficult because the resulting datasets are highly unbalanced. All documents in train/valid/test come from Reuters-21578. Data in the five sets do not overlap. Data statistics can be found in Table 7.

- **TrainingSet** We choose articles in the trianset of Reuters-ModApte belonging to m topics.
- **ID-ValidSet** We choose articles in the valid set of Reuters-ModApte belonging to m topics.
- **ID-TestSet** We choose articles in the test set of Reuters-ModApte belonging to m topics.
- **OOD-ValidSet** We choose articles in the valid set of Reuters-ModApte belonging to n topics.
- **OOD-TestSet** We choose articles in the test set of Reuters-ModApte belonging to n topics.

AGNews-FL The dataset is composed of data from AGNews and additional articles from the AG Corpus. In this setting, the training and ID data are from the 4 classes ("World", "Sports", "Business", "Sci/Tech") in AGNews, and the OOD data are from another 4 classes ("U.S.", "Europe", "Italia", "Software and Development") in AG Corpus. It is noteworthy that these two sets of labels are similar in semantics, e.g., "U.S." to "World", "Europe" to "Sports" and "Software and Development" to "Sci/Tech". This makes the task more challenging than AGNews-FM, which will be introduced below. Data in the five sets do not overlap.

- **TrainingSet** We use 116,000 articles from the trainset in AG-News belonging to 4 classes. Each class contains 29,000 articles.
- **ID-ValidSet** We use 4,000 articles from the trainset in AG-News. Each class has 1,000 articles.
- **ID-TestSet** We use 4,000 articles from the testset in AG-News. Each class has 1,000 articles.

[10] http://kdd.ics.uci.edu/databases/reuters21578/README.txt

Train	**Acq**	**Corn**	**Crude**	**Earn**	**Grain**	**Interest**	**Money-fx**	**Ship**	**Trade**	**Wheat**
Reuters	1615	175	383	2817	422	343	518	187	356	206
Reuters-9K-1L	1615	175	383	2817	422	343	518	N/A	356	206
Reuters-6K-4L	1615	175	N/A	2817	422	343	518	N/A	N/A	N/A
Reuters-5K-5L	1615	N/A	N/A	2817	422	343	518	N/A	N/A	N/A
Reuters-3K-7L	1615	N/A	N/A	2817	N/A	N/A	518	N/A	N/A	N/A
Reuters-2K-8L	1615	N/A	N/A	2817	N/A	N/A	N/A	N/A	N/A	N/A

ID Test	**Acq**	**Corn**	**Crude**	**Earn**	**Grain**	**Interest**	**Money-fx**	**Ship**	**Trade**	**Wheat**
Reuters	719	56	189	1087	149	131	179	89	117	71
Reuters-9K-1L	719	56	189	1087	149	131	179	N/A	117	71
Reuters-6K-4L	719	56	N/A	1087	149	131	179	N/A	N/A	N/A
Reuters-5K-5L	719	N/A	N/A	1087	149	131	179	N/A	N/A	N/A
Reuters-3K-7L	719	N/A	N/A	1087	N/A	N/A	179	N/A	N/A	N/A
Reuters-2K-8L	719	N/A	N/A	1087	N/A	N/A	N/A	N/A	N/A	N/A

OOD Test	**Acq**	**Corn**	**Crude**	**Earn**	**Grain**	**Interest**	**Money-fx**	**Ship**	**Trade**	**Wheat**
Reuters	719	56	189	1087	149	131	179	89	117	71
Reuters-9K-1L	N/A	N/A	N/A	N/A	N/A	N/A	N/A	89	N/A	N/A
Reuters-6K-4L	N/A	N/A	189	N/A	N/A	N/A	N/A	89	117	71
Reuters-5K-5L	N/A	56	189	N/A	149	131	179	N/A	N/A	N/A
Reuters-3K-7L	N/A	56	189	N/A	149	131	N/A	89	117	71
Reuters-2K-8L	N/A	56	189	N/A	149	131	179	89	117	71

Table 7: Data statistics for Reuters-mK-nK datasets.

- **OOD-ValidSet** We assemble titles and description fields of articles in AG Corpus from another 4 classes different from AG-News. There are 4,000 articles and 1,000 articles per class.
- **OOD-TestSet** We assemble titles and description fields of articles in AG Corpus from another 4 classes different from AG-News. There are 4,000 articles and 1,000 articles per class.

AGNews-FM The dataset is composed of data from AGNews and additional articles from the AG Corpus. In this setting, the training and ID data are from the 4 classes ("World", "Sports", "Business", "Sci/Tech") in AGNews, and the OOD data are from another 4 classes ("Entertainment", "Health", "Top Stories", "Music Feeds") in AG Corpus. This dataset is easier than AGNews-FL because the OOD labels are more distinct from the ID labels regarding the label semantics. Data in the five sets do not overlap.

- **TrainingSet** We use 116,000 articles from the trainset in AG-News belonging to 4 classes. Each class contains 29,000 articles.
- **ID-ValidSet** We use 4,000 articles from the trainset in AG-News. Each class has 1,000 articles.
- **ID-TestSet** We use 4,000 articles from the testset in AG-News. Each class has 1,000 articles.
- **OOD-ValidSet** We assemble titles and description fields of articles in AG Corpus from another 4 classes different from AG-News. There are 4,000 articles and 1,000 articles per class.
- **OOD-TestSet** We assemble titles and description fields of articles in AG Corpus from another 4 classes different from AG-News. There are 4,000 articles and 1,000 articles per class.

Yahoo!Answers-FM This dataset is modified from the Yahoo!Answers dataset. We use five topic articles ("Health", "Science & Mathematics", "Sports", "Entertainment & Music", "Business & Finance") for the training and ID data and use the other five unseen topics ("Society & Culture", "Education & Reference", "Computers & Internet", "Family & Relationships", "Politics & Government") for the OOD data. Data in the five sets do not overlap.

- **TrainingSet** We use 680,000 examples belonging to five categories in Yahoo!Answers, 136,000 samples per class.

- **ID-ValidSet** We use 20,000 examples belonging to five categories in Yahoo!Answers. 4,000 samples per class.
- **ID-TestSet** We use 25,000 examples belonging to five categories in Yahoo!Answers. 5,000 samples per class.
- **OOD-ValidSet** The data are from another five categories in Yahoo!Answers. The OOD-ValidSet contains 20,000 articles with 4,000 per class.
- **OOD-TestSet** The data are from another five categories in Yahoo!Answers. The OOD-TestSet contains 25,000 articles with 5,000 per class.