Transfer Learning for Text Classification via Model Risk Analysis

Yujie Sun[1], Chuyi Fan[1], Qun Chen[1,2],

[1]School of Software, Northwestern Polytechnical University, Xi'an, China
[2]School of Computer Science, Northwestern Polytechnical University, Xi'an, China
Correspondence: chenbenben@nwpu.edu.cn

Abstract

It has been well recognized that text classification can be satisfactorily performed by Deep Neural Network (DNN) models, provided that there are sufficient in-distribution training data. However, in the presence of distribution drift, a well trained DNN model may not perform well on a new dataset even though class labels are aligned between training and target datasets. To alleviate this limitation, we propose a novel approach based on model risk analysis to adapt a pre-trained DNN model towards a new dataset given only a small set of representative data. We first present a solution of model risk analysis for text classification, which can effectively quantify misprediction risk of a classifier on a dataset. Built upon the existing framework of **LearnRisk**, the proposed solution, denoted by **LearnRisk-TC**, first generates interpretable risk features, then constructs a risk model by aggregating these features, and finally trains the risk model on a small set of labeled data. Furthermore, we present a transfer learning solution based on model risk analysis, which can effectively fine-tune a pre-trained model toward a target dataset by minimizing its misprediction risk. We have conducted extensive experiments on real datasets. Our experimental results show that the proposed solution performs considerably better than the existing alternative approaches. By using text classification as a test case, we demonstrate the potential applicability of risk-based transfer learning to various challenging NLP tasks. Our codes are available at https://github.com/syjcomputer/LRTC.

1 Introduction

As a very important task in natural language processing, text classification aims to categorize a given text into multiple groups based on its contents. Text classification, including the more specific tasks of sentiment analysis (Abbas et al., 2019), news categorization (Chen et al., 2022) and topic classification (Pappagari et al., 2019), have been extensively studied in the literature (Minaee et al., 2021). With the emergence of large language models (e.g., Bert (Devlin et al., 2019), BAE (Garg and Ramakrishnan, 2020) and BertGcn (Lin et al., 2021)), the research community has experienced a considerable shift towards how to adapt these models to the task.

It has been well recognized that DNN models can usually perform well on text classification, provided that there are sufficient in-distribution training data (Chen et al., 2022; Devlin et al., 2019). However, in the presence of distribution drift, a well trained DNN model may not perform well on a new dataset even though class labels are aligned between training and target datasets . As other mainstream machine learning models, the efficacy of DNN models depends on i.i.d (Identically and Independently Distributed) assumption. Unfortunately, in real applications, it is usually very labor-intensive thus prohibitive to retrieve a sufficient amount of in-distribution labeled data. In these scenarios, it is desirable that a pre-trained model can be easily adapted to a new dataset with only a small amount of additional labeled data. It is noteworthy that the community has presented many transfer learning approaches, targeting many tasks including but not limited to text classification (Pan and Yang, 2009; Ying et al., 2018; Zhuang et al., 2021). These existing work mainly focused on how to mine the knowledge from training data and apply them on target data. However, most of them can not effectively adapt a model towards a new dataset by its particular characteristics. Therefore, transfer learning generally remains very challenging, with text classification being no exception.

On the other hand, we have observed that AI model risk analysis has garnered much attention in recent years due to the concern on AI model's misbehavior (Hendrycks and Gimpel, 2017; Hendrycks et al., 2019; Jiang et al., 2018). The purpose of model risk analysis is to quantify a

Findings of the Association for Computational Linguistics: EACL 2024, pages 2814–2825
November 12-16, 2024 ©2024 Association for Computational Linguistics

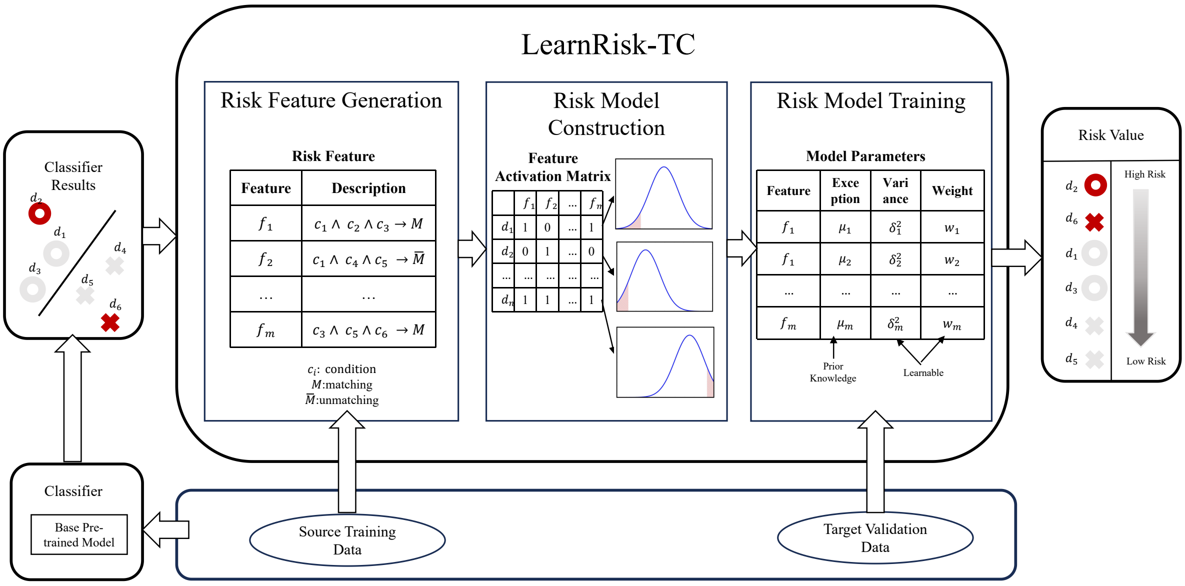


Figure 1: The Framework of the **LearnRisk-TC** solution:1) risk feature generation; 2) risk model construction by aggregating the distributions of risk features; 3) risk model training by a learn-to-rank objective, which ensures mispredictions be ranked before correct predictions.

model's misprediction risk on a dataset, and then leverage the results of risk analysis for adaptive model fine-tuning (Zhang et al., 2022). Since risk analysis can be potentially effectively performed with only a small amount of labeled data, it provides a viable technical roadmap for transfer learning. In this paper, we first present a solution of model risk analysis for text classification, and then propose a corresponding risk-based transfer learning approach, which can effectively adapt a DNN model towards a target dataset by its particular characteristics.

Specifically, because the risk analysis framework of LearnRisk, which was originally proposed for the task of entity resolution (Chen et al., 2020), is more interpretable and accurate than the existing alternatives, we have built the solution of risk analysis for text classification based on LearnRisk. We have sketched the solution of risk analysis, which is denoted by **LearnRisk-TC**, in Figure 1. It first generates interpretable risk features, then constructs a risk model by aggregating these features, and finally trains the risk model using a small set of representative data. Furthermore, we present a corresponding solution of transfer learning based on risk analysis. The overall solution of transfer learning is shown in Figure 2. It consists of two training phases. The first phase is the same as the process of traditional model training, which trains a model based on labeled data from a source. In the second phase, it furthermore fine-tunes the model on a target dataset by minimizing its misprediction risk.

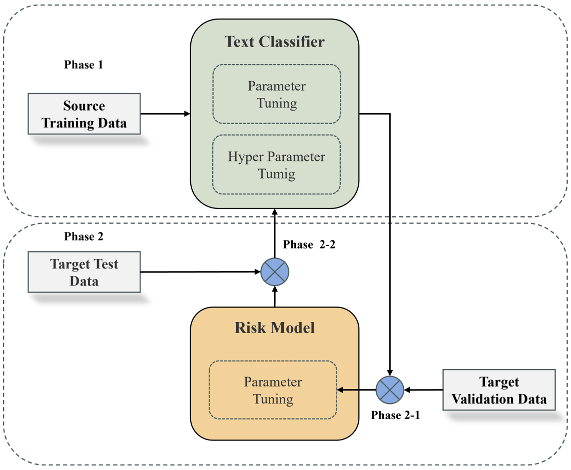


Figure 2: Risk-based Transfer Learning.

Our main contributions in this paper can be summarized as follows:

- We propose a novel solution of interpretable model risk analysis for text classification, which can accurately quantify a model's misprediction risk on a given dataset;
- We propose a transfer learning solution for

text classification based on risk analysis, which can effectively adapt a model towards a target datasetworkload by its particular characteristics;

- We empirically validate the efficacy of the proposed solutions on real datasets. Our extensive experiments show that in the scenario of distribution drift, provided with only a small set of representative data, the proposed solution of risk analysis can more accurately identify mispredictions than the existing alternatives. Furthermore, the proposed solution of transfer learning similarly outperforms the existing alternatives by considerable margins.

2 Related Work

In this section, we review related work from the orthogonal perspectives of text classification, risk analysis and transfer learning.

Text Classification. Traditional solutions for text classification were constructed based on statistics and machine learning, e.g., KNN (Trstenjak et al., 2014), Naïve Bayes (Abbas et al., 2019) and TF-IDF (Jiang et al., 2021). The hybrid approach of fusing the KNN and the TF-IDF was first explored in 2014 (Trstenjak et al., 2014). The approach that combined KNN and clustering, which could considerably reduce similarity computation, was first proposed in 2014 (Jiang et al., 2012). The Naïve Bayes method was also a popular approach for text classification (Abbas et al., 2019).

With the emergence of deep learning, various DNN models have been proposed for text classification. In recent years, the research community has experienced a considerable shift towards pre-trained large language models, the most prominent among which is the Bert model (Devlin et al., 2019). In the following work, the Bert model has been extended into multiple variations, including BAE (Garg and Ramakrishnan, 2020) and BertGCN (Lin et al., 2021). Additionally, contrastive learning and adversarial learning have been employed for text classification to facilitate labeling with fewer labeled samples. UST introduced Bayesian networks for uncertainty estimations to improve the classification performance on few-label tasks (Mukherjee and Awadallah, 2020). NSP-BERT used templates to perform inference on test samples using Masked Language Modeling (MLM) (Sun et al., 2022). More recently, Large Language Models (LLMs) such as LLaM and Alpaca-7B were tuned using the techniques such as Recurrently Ensembling Base Learners (RGPT) (Zhang et al., 2024b) or based on Instruction through Contrastive Self-training (Zhang et al., 2024a), demonstrating impressive performance on zero-shot text classification tasks. It is noteworthy that despite their efficacy, these LLMs often require significant computational resources.

Risk Analysis. It has been empirically shown that even a well-trained model usually does not perform satisfactorily if employed in real scenarios. Therefore, the study of risk analysis has drawn much attention in recent years (Hendrycks and Gimpel, 2017; Hendrycks et al., 2019; Jiang et al., 2018; Chen et al., 2020; Nafa et al., 2024a). The authors of (Hendrycks and Gimpel, 2017) proposed a simple baseline that employed the probabilities from softmax distributions. The authors of (Jiang et al., 2018) proposed a new metric called TrustScore to measure the agreement between the classifier and a modified nearest-neighbor classifier on test examples. More recently, a more interpretable framework of LearnRisk was proposed for the task of entity resolution (Chen et al., 2020, 2018). Since its introduction, the framework has also been extended to handle the tasks of entity disambiguation (Nafa et al., 2024b) and network intrusion detection (Zhang et al., 2022). In these papers, the authors also demonstrated that model risk analysis could be leveraged to effectively facilitate adaptive deep learning. In this paper, we have built the solutions of risk analysis and risk-based transfer learning for text classification based on LearnRisk.

Transfer Learning. As a technique of adapting pre-trained models to new datasets, transfer learning can effectively reduce the need for large amounts of training data, improving model generalization and accelerating model training (Zhuang et al., 2021). Broadly speaking, the work on transfer learning includes more specific sub-areas, e.g., subpopulation shift (Alshaer et al., 2021), domain generalization (Li et al., 2018) and domain adaptation (Farahani et al., 2021). In this paper, we focus on domain adaptation, which aims to adapt a model pre-trained on a source dataset to a target dataset with a different distribution.

In terms of NLP tasks, with the emergence of pre-trained large language models, transfer learning has become a standard approach to adapt these models to downstream tasks such as text classification (Zhuang et al., 2021). The two mainstream approaches for transfer learning in NLP are direct model fine-tuning and feature freezing (Lee et al., 2019). The first approach directly fine-tunes a pre-trained model on new tasks with a few additional epochs and a small learning rate. This approach can lead to high additional costs and may ignore the similarity between the source and target datasets. The approach of feature freezing instead keeps the earlier layers of a pre-trained model unchanged while only training the later layers, assuming that the early layers capture general features that are transferable across tasks. Since determining the proper number of layers to freeze is challenging, and researchers have proposed various methods to optimize the number of layers to be freezed (Liu et al., 2021; Lee et al., 2019). However, the assumption of feature freezing that some layers retain similarity while others are only related to the source dataset may be problematic in real scenarios.

3 Task Statement

In this paper, we consider text classification as a multi-label classification problem. Given a text, a classifier needs to output a label for the text, which can accurately summarize its topic. As usual, we use accuracy as the evaluation metric.

Formally, we define the task of text classification by:

Definition 1. ***[Task of Text Classification.]*** *Given a dataset of text classification, D, consisting of D_s, D_v and D_t, where D_s, D_v and D_t denote the sets of training data, validation data and test data respectively, the task aims to learn an optimal classifier, $g(w_*)$ based on D such that the performance of $g(w_*)$ on D_t as measured by the metric of accuracy is maximized.*

It is noteworthy that in this paper, we focus on the scenario of transfer learning, where the validation and test data, D_v and D_t, come from a source other than that of D_s, and there is distribution drift between these two text sources. We also suppose that the validation dataset, D_v, which are representatives of D_t, contains only a limited amount of labeled data. Otherwise, in real scenarios, the labeled data in D_v can be directly used to fine-tune a pre-trained model, making transfer learning unnecessary.

4 Risk Analysis Solution: LearnRisk-TC

In this section, we first briefly introduce the existing framework of LearnRisk, and then present the LearnRisk-TC solution for text classification.

4.1 The Framework of LearnRisk

As shown in Figure 1, the general LearnRisk framework mainly consists of three steps: risk feature generation, risk model construction and risk model training:

1) Risk Feature Generation: the first step is to construct risk metrics and then automatically generate interpretable risk features based on risk metrics. It needs to ensure the generated risk features are discriminative, i.e., each rule is highly indicative of one class label over others. Furthermore, their validity needs to span over a considerable subpopulation of the workload. In LearnRisk, risk features are usually represented by one-sided decision rules. Unlike the traditional labeling functions, a risk rule concentrates solely on a single class. Consequently, a risk feature acts as an indicator of the case where a classifier's prediction goes against the knowledge embedded in it. LearnRisk usually employs the technique of one-sided decision trees to generate high-quality risk features.

An illustrative example of risk feature is:

$$dist(d_i, C_j) < 0.134 \wedge knn_5(d_i, C_j) \geq 4 \rightarrow d_i \in C_j, \tag{1}$$

where d_i denotes a document text, C_j denotes a text class, $dist(d_i, C_j)$ denotes the distance between d_i and the centroid of C_j in an embedding space, $knn_5(d_i, C_j)$ denotes the number of documents belonging to the C_j among the 5 nearest neighbors of d_i . According to Eq. 1, if a document d_i satisfies both conditions specified in the rule, it belongs to C_j.

2) Risk Model Construction: the second step of risk model construction, LearnRisk constructs a risk model for estimating classifier risk using risk features generated in the first step. Inspired by investment theory, it models the distribution of an instance belonging to a certain class by aggregating the distributions of each risk feature.

Specifically, given a class, C_i, and its set of m risk features, $F = \{f_1, f_2, \ldots, f_m\}$, we use $u_F = [u_{f_1}, u_{f_2}, \ldots, u_{f_m}]^T$ and $\delta_F^2 = [\delta_{f_1}^2, \delta_{f_2}^2, \ldots, \delta_{f_m}^2]^T$ to denote the corresponding mean and variance for each risk feature. We also denote their corresponding feature weight vector by $w = [w_1, w_2, \ldots, w_m]^T$. Then we can calculate the cumulative distribution of the class C_i with probability distributions of risk features. The mean and variance of distribution of the class C_i can be represented by:

$$u_i = z_i \cdot (w \circ u_F) \tag{2}$$

$$\delta_i^2 = z_i \cdot (w \circ \sigma_F^2) \tag{3}$$

where $\circ$ represents the element-wise product and $\cdot$ represents matrix multiplication. z_i is a one-hot feature vector. Specifically, $z_i = [z_{i1}, z_{i2}, \dots, z_{im}]$, where $z_{ij} = 1$ if d_i has the $j-th$ feature, otherwise $z_{ij} = 0$.

LearnRisk typically uses the metric of Value-at-Risk (VaR) (Tardivo, 2002) to estimate the misprediction risk. The metric of VaR represents the maximum loss that may be incurred, excluding the worst-case scenario with a total occurrence probability of $1 - \theta$. θ is a confidence level.

3) Risk Model Training: in the final step of risk model training, LearnRisk trains a risk model on labeled representative data through optimizing a learn-to-rank objective. It usually considers the expectations (u_i) as prior knowledge and estimate them by labeled training data, but adjusts both the variances (δ_i^2) and weights (w_i) of risk features for model adaptation. After the training process, the risk model can be applied to evaluate the misprediction risk on unlabeled instances, as determined by a classifier.

4.2 Solution: Risk Feature Generation

To enable model risk analysis for text classification, we first extract risk metrics, and then leverage one-sided decision trees to generate interpretable risk features. In LearnRisk-TC, we extract risk metrics based on linguistic statistics and DNN models respectively.

1) Extraction of statistics-based risk metrics: to extract statistics-based risk metrics, we first extract the top-K representative words for each category, which constitute its feature dictionary, and then quantify a document's relevance to a category by the hits of feature words in the document. Intuitively speaking, we select the words that are heavily present in a category but not in others, as the feature words of the category. It can be observed that if a document has a high hit rate of feature words in a certain category, it is more likely to belong to this category.

Specifically, our solution extracts feature words by a hybrid metric consisting of both an improved TF-IDF measure (Khan et al., 2021; Aljedaani et al., 2022) and a Chi-square measure (Kumar et al., 2021; Alshaer et al., 2021). The hybrid metric can be represented by:

$$\begin{aligned} TFIDF - CHI = p \times CHI_{new} \\ + (1-p) \times TF - IDF_{\text{new}} \end{aligned} \tag{4}$$

where p denotes a trade-off weight between two measures. In practical implementation, we suggest to set the value of p as a value below 0.5, e.g., [0.1,0.3].

2) Extraction of DNN-based risk metrics: since the success of DNN on NLP tasks depends on embedding representations, a document's vector representation is highly indicative of its relevance to a category. Therefore, we also leverage DNN-based vector representations to generate indicative risk metrics. Specifically, we use labeled training data to fine-tune the mainstream language models, then extract both pooler and dense layer outputs as vector representations, and finally construct corresponding risk metrics based on similarity/distance measurement. The pooling layer provides a hidden state that serves as a sentence-level representation, while the output of the dense layer is used for specific tasks.

Specifically, we extract two types of embedding-based risk metrics as follows:

- **KNN.** Given a document d_i, we count the number of its k nearest neighbors in each category. Intuitively speaking, if all the k nearest neighbors belong to a certain category, then d_i is very likely to belong to this category. In practical implementation, we can set $k$at different values to generate multiple KNN risk metrics.

- **Class Centroid Distance (CCD).** Also based on vector representations, we estimate a document's distance to class centroids. It can be observed that the smaller the distance is, the more likely the document belongs to the corresponding category.

Our implementation used two DNN models, i.e., Bert and TextCNN, to extract embedding-based risk metrics separately. It is noteworthy that other DNN models can be similarly applied to extract embedding-based risk metrics. Our experimental results show that by using Bert and TextCNN, both of which are mainstream but preliminary language models, our proposed approach can effectively outperform other more advanced DNN models.

Finally, provided with the extracted risk metrics, our solution generates risk features by one-side decision trees as presented in (Chen et al., 2020). One-side decision trees ensure that most documents in D_i have the same label, but the other document in D can have mixed labels. The label purity of a one-sided partition is measured by the metric of the one-sided Gini index as follows:

$$\begin{aligned} G(D) = min(&\frac{\lambda}{|D_l|} + (1-\lambda) \times G(D_l), \\ &\frac{\lambda}{|D_r|} + (1-\lambda) \times G(D_r)) \end{aligned} \tag{5}$$

where λ denotes a weight parameter to balance the influence of set size and label impurity, $G(D_l)$ and $G(D_r)$ denote the Gini values of two subsets.

It is noteworthy that our solution generates a distinct set of risk features for each class. Specifically, given a class and labeled training data, it first labels a text as 1 or 0, with 1 means belonging to the class while 0 indicating otherwise. Then, it leverages one-sided decision trees to generate risk features for this class.

4.3 Solution: Risk Model Construction and Training

Our solution constructs a separate risk model for each text class. As in the previous work (Chen et al., 2020), we employ the metric of Value at Risk (VaR) to quantify misclassification risk. Assuming the probability that d_i is classified as C_j is p_i, we denote the misclassification probability of d_i by $1 - p_i$ and the inverse of its cumulative distribution function by $F_i^{-1}(\cdot)$. Given the confidence level of θ, the VaR risk of d_i can be calculated by:

$$VaR(d_i) = 1 - F_i^{-1}(1-\theta; u_i, \delta_i^2) \tag{6}$$

where u_i and δ_i^2 denote the parameters of $F_i^{-1}(1-\theta)$.

As usual, we train the risk model on labeled representative data by a learning-to-rank objective (Burges et al., 2005). The aim of risk model is to rank the documents with high mislabeling risk before the documents with low mislabeling risk. The parameters that should be learned include the weights of risk features, w_i, and their distribution variances, δ^2. The expectations (u_i) are instead considered as prior knowledge and estimated by labeled training data. We employ the logistic function to map a risk value to a posterior probability by:

$$p_{ij} = \frac{e^{(var_i - var_j)}}{1 + e^{(var_i - var_j)}}, \tag{7}$$

and we define its target probability by:

$$\bar{p}_{ij} = 0.5 \times (1 + \hat{g}_i - \hat{g}_j) \tag{8}$$

where $\hat{g}_i$ and $\hat{g}_j$ denote risk labels of documents, with 0 and 1 mean being correctly labeled and mislabeled respectively. Finally, based on the definitions of posterior probability and its target probability, we define the cross-entropy loss function of risk model training by:

$$\begin{aligned} L(D) = \sum_{d_i, d_j \in D} \Big[&-\bar{p}_{ij} \times \log(p_{ij}) \\ &- (1-\bar{p}_{ij}) \times \log(1-\bar{p}_{ij}) \Big] \end{aligned} \tag{9}$$

5 Transfer Learning Solution

To enable transfer learning, we introduce an adaptive deep learning approach for text classification. As shown in Figure. 2, it consists of two phases of training. In the first phase, it trains a base DNN model (e.g., Bert, in our implementation) based on D_s by the cross-entropy loss function in the traditional way. In the second phase, it further fine-tunes the model on D_t by minimizing misprediction risk. Specifically, it iteratively performs: 1) using LearnRisk-TC to learn a risk model for each class based on a trained classifier and labeled validation data of D_v; 2) fine-tuning the classifier by minimizing its misprediction risk upon the target dataset of D_t.

Now, we discuss how to minimize the misprediction risk. For each text in D_t, our solution estimates a probability distribution of each class based on its corresponding risk model. Then, it selects the class with with the highest mean (or u) as its predicted label. Finally, it uses the text and its predicted label to fine-tune a classifier by the loss function of

$$\begin{aligned} &L_{\text{test}}^{\text{risk}}(w) = \\ &\frac{1}{n_{\text{s}}} \sum_{i=1}^{n_{\text{s}}} \left[-\left(1 - \text{VaR}^{+}(d_i)\right) \log\left(g\left(x_i^{\text{s}}, w\right)\right) \right] \end{aligned} \tag{10}$$

where n_{s} denotes the number of test data in D_B, $VaR^{+}(d_i)$ denotes the risk value if d_i is labeled as its predicted class, $VaR^{-}(d_i)$ denotes the risk value if d_i is labeled as a class other than its predicted class.

We have sketched the whole process in Algorithm 1. The first step trains the classifier with

Algorithm 1 Risk-based Adaptive Training

```
Input: $D_s$, $D_v$ and $D_t$, a classifier $g(w)$
Output: A learned classifier $g(w_*)$.
$w_0 \leftarrow$ initialize $w$ with random values
for $k = 0$ to $m - 1$ do
    $w_{k+1} \leftarrow w_k - \alpha \times \nabla_{w_k} L_{\text{train}}(w_k)$
end for
Select the best model $g(w_*)$
$w_m \leftarrow w_*$
for $k = 0$ to $n - 1$ do
    Update the risk models based on $D_v$
    $w_{k+1} \leftarrow w_k - \alpha \times \nabla_{w_k} L^{\text{risk}}_{\text{test}}(w)$
    Update $g(w_k)$ based on the trained risk models and $D_t$
end for
Select the last model $g(w_*) \leftarrow g(w_{k+1})$
return g($w_*$)
```

D_s and selecting the best classifier, then we use the selected classifier $g(w_*)$ and D_v to train risk models. Finally, we use the learned risk models to fine-tune our classifier and select the best model, $g(w_*)$, based on D_t .

6 Empirical Evaluation

In this section, we evaluate the performance of our proposed solutions by a comparative study on real benchmark datasets. Subsection 6.1 describes our experimental setup. Subsection 6.2 presents the evaluation results of risk analysis. Subsection 6.3 presents the evaluation results of transfer learning. Finally, in Subsection 6.4, we evaluate the performance sensitivity of the proposed solutions w.r.t the size of validation data, which serve as the representatives of a target dataset.

6.1 Experiment Setup

Table 1: Statistics of Evaluation Datasets

Dataset Pair	Classes	Train	Validation	Test
20News-BBC	4	4763	310	915
20News-AgNews	3	6062	364	1140
AgNews-BBC	4	3192	409	559
BBC-20News	4	480	311	7981
AgNews-20News	3	2736	420	6525
BBC-AgNews	4	768	425	2280

We have used three benchmark news datasets in our experiments, which included 20News [1], BBC2 [2] and AgNews (Zhang et al., 2015). To simulate the scenario of distribution drift, our experiments suppose that training data come from a dataset while validation and test data come from another dataset. We have summarized the statistics of our test datasets in Table 1. In the table, the dataset name of 20news-BBC means that train data come from the 20News dataset but validation and test data come from the BBC dataset. It is noteworthy that we intentionally limit the number of labeled validation data, i.e., less than 500, to simulate the scenario of transfer learning in real applications, where labeled representatives are usually not readily available.

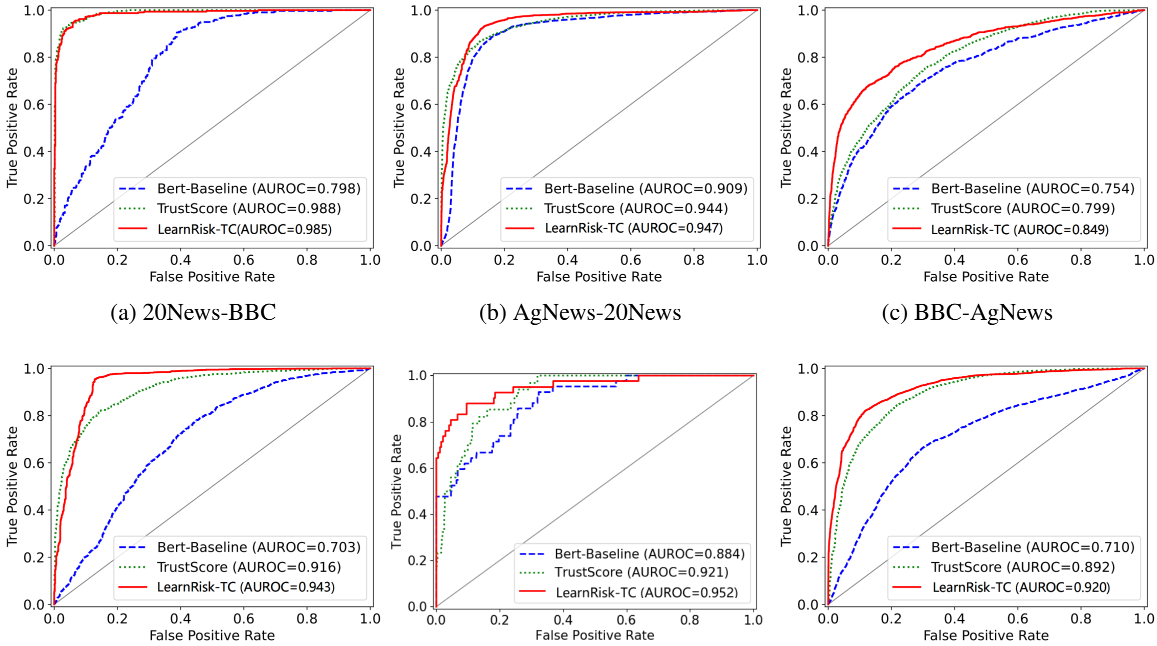
For risk analysis, we compare our proposed solution, denoted by LearnRisk-TC, with two mainstream alternatives: 1) **Baseline**: it directly measures misprediction risk by the output of a classifier, the baseline model of Bert in our implementation, or (1-p), where p is the label probability given by the model; 2) **TrustScore** (Jiang et al., 2018): it is a distance-based risk measure. Given a document d_i labeled as C_j, the metric of TrustScore is defined by the ratio of the distance from the nearest class except C_j for the data to the distance from the data to the alpha high-density set of C_j. The TrustScore metric implies that the closer d_i is to its prediction class centroid, the lower risk it has.

For the evaluation of risk-based transfer learning, we compare our solution with the existing mainstream pre-trained models proposed for text classification. For fair comparison, we train these models in two phases, the first phase based on training data from a source dataset and the second one based on validation data from a target dataset. Note that in the second phase, we train models by direct fine-tuning, because direct fine-tuning performs overall better than feature freezing on the test datasets. The compared pre-trained models and their model sizes are as follows: 1) **Bert**(Devlin et al., 2019) (110M parameters); 2) **Roberta**(Liu et al., 2019) (125M parameters); 3) **XLNet**(Yang et al., 2019) (110M parameters); 4) **TextCNN**(Kim, 2014) (4.6M parameters); 5) **BertGCN**(Lin et al., 2021) (110M+ parameters); 6) **npc-gizp**(Jiang et al., 2023); 7) **CAT**(Pan et al., 2022) (110M+ parameters). Note that npc-gzip relies on a compression algorithm and does not have a specific parameter count. In addition, we have also compared our solution with two deep learning methods proposed for text classification in the scenario of limited labeled training data: **UST**(Mukherjee and Awadallah, 2020) (110M parameters) and **NSP-BERT** (Sun et al., 2022) (110M parameters).

We have implemented the solution of

[1] http://qwone.com/ jason/20Newsgroups/

[2] http://mlg.ucd.ie/datasets/tbbc.html



(a) 20News-BBC

(b) AgNews-20News

(c) BBC-AgNews

(d) 20News-AgNews

(e) AgNews-BBC

(f) BBC-20News

Figure 3: Evaluation Results of Risk Analysis

LearnRisk-TC based on the open-sourced implementation of LearnRisk [3].In the extraction of statistics-based risk metrics, we set the parameter of p, which denotes the trade-off weight between the measures of TF-IDF and CHI, at 0.1. In the extraction of DNN-based risk metrics, we set the parameter of k, which denotes the number of k-nearest neighbors at 5 and 8. In the implementation of risk feature generation, we set the maximum depth of decision trees at 3, and the Gini theresholds for class matching and unmatching at 0.3 and 0.03 respectively. In the implementation of risk-based adaptive training, we set the learning rate at a small value of 5e-5, and the iteration number of adaptive fune-tuning at 5, i.e., we fine-tune the classifier with 5 additional iterations in the second phase. Our experiments have shown that after 5 iterations, the performance of adaptive model becomes stable. Our experiments have been conducted on a machine with 2 NVIDIA RTX 3090 GPUs, each with 24 GB of memory.

6.2 Evaluation Results of Risk Analysis

As usual, we use the metric of AUROC to evaluate the performance of risk analysis. The detailed evaluation results have been presented in Figure 3. It can be observed that the Baseline performs the worst in all the datasets, and the margins on some of them, e.g., 20News-BBC and BBC-20News, are very considerable. Over-confidence is a common issue for DNN models, outputing high-confidence probabilities even when making wrong predictions. Our experimental results demonstrates that in terms of over-confidence, DNN models for text classification are no exception. In comparison, the proposed LearnRisk-TC achieves the best performance on five out of the six datasets, with the exception of 20news-BBC, where its performance is slightly worse than that of TrustScore (0.985 vs 0.988). It is also noteworthy that on 5 out of the 6 datasets, LearnRisk-TC achieves high AUROC values, over 0.9. Our experimental results clearly demonstrate the performance advantage of LearnRisk-TC over the alternatives of Baseline and TrustScore.

6.3 Evaluation Results of Transfer Learning

We have presented the detailed evaluation results of transfer learning in Table 2. It can be observed that LearnRisk-TC achieves the best performance on 5 out of the 6 datasets, with the exception of BBC-20News , where it performs slightly worse than CAT and NSP, 68.19% vs 69.26% and 70.13%. It is interesting to point out that the variations of Bert, Roberta and XLnet, perform even worse then the baseline model of Bert, meaning that additional

[3] https://chenbenben.org.cn/RiskER/RiskER.zip

Table 2: Evaluation Results of Transfer Learning

Methods	datasets					
	20News-AgNews	AgNews-20News	AgNews-BBC	BBC-AgNews	BBC-20News	20News-BBC
Bert	91.2	83.1	89.48	78.92	62.68	88.75
Roberta	77.38	79.47	65.87	75.06	54.20	61.8
XLnet	82.87	86.23	82.77	63.05	58.87	81.06
TextCNN	43.78	61.05	64.39	41.23	54.64	38.79
BertGCN	36.06	54.97	35.36	36.38	52.4	40.04
BERT+CAT	31.2	50.39	36.42	35.61	30.7	44.5
npc-gzip	43.51	79.05	59.22	32.24	54.76	46.45
BERT+UST	90.96	83.4	92.12	73.13	69.26	70.93
NSP	81.94	85.87	77.64	75.79	**70.13**	91.69
LearnRisk-TC	**92.11**	**86.32**	**93.92**	**78.94**	68.19	**93.22**

twisting of Bert model may sacrifice its generalizability capability in the more challenging scenario of limited in-distribution training data. BertGCN and UST, with more training and model twisting than others, even perform considerably worse on many workloads, including BBC-AgNews, BBC-20News and 20News-BBC.

It can be observed that UST and NSP-BERT overall perform better than the models constructed without considering distribution drift. These results validate the efficacy of their designed mechanisms of transfer learning. However, LearnRisk-TC still manages to outperform them, with considerable margins on some datasets, e.g., around 20% improvement over UST on 20News-BBC and around 10% improvement over NSP on 20News-AgNews. These experimental results show that LearnRisk-TC can better leverage a small set of labeled representatives to adapt a DNN towards a target dataset.

6.4 Robustness Evaluation w.r.t Size of Validation Data

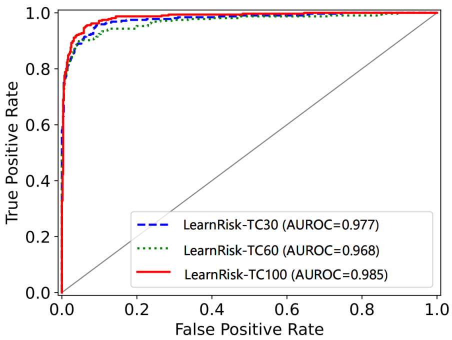


Figure 4: Robustness Evaluation of Risk Analysis

This subsection evaluates the performance sensitivity of our proposed solutions w.r.t the size of validation data. We present the evaluation results on the dataset of 20News-BBC. The results on other datasets are similar, thus omitted here due to space limit. We reduce the size of validation data by selecting only a portion of labeled data, i.e., 30% and 60%.

Table 3: Robustness Evaluation of Transfer Learning

dataset	Validation Size	Accuracy
20News-BBC	103	93.17
	206	93.16
	310	93.22

The detailed evaluation results of risk analysis and transfer learning have been presented in Figure 4 and Table 3 respectively. It can be observed that in terms of both AUROC and classification accuracy, the performance of LearnRisk-TC only fluctuates marginally as the size of validation data decreases. These experimental results clearly demonstrate that LearnRisk-TC can effectively adapt a DNN model towards a target dataset by leveraging only a small set of representative data.

7 Conclusion

In this paper, we have proposed a novel solution of model risk analysis for text classification, and a corresponding risk-based solution of transfer learning to adapt a DNN model towards a target dataset by using only a small set of representative data. Our experiments on real datasets have validated the efficacy of the proposed solutions. On future work, it is interesting to point out that the proposed risk-based approach of transfer learning can be potentially applied to other NLP tasks. However, the detailed technical solutions require further investigation.

Acknowledgments

The work was supported by the National Natural Science Foundation of China (Grant No. 62172335 and 61972317), the National Key Research and Development Program of China (2023YFB4503600).

Limitations

Our work has the following limitations:

- Our proposed solutions have been empirically shown to be effective on the open-sourced news classification datasets. However, future research needs to be conducted on more diverse datasets.

- Our current solutions extract both keyword-based and DNN embedding-based risk metrics. Unfortunately, the interpretability of embedding-based risk metrics, i.e., KNN and CCD metrics, is low due to the general poor interpretability of DNN models. It is interesting to investigate how to construct more interpretable risk features for text classification.

- Our work didn't perform risk analysis for AIGC large language models (LLMs), e.g., ChatGPT-4 and Gemini, because based on recent tests by the research community, they can't considerably outperform the mainstream classification language models on text classification. However, as AIGC models evolve, it is interesting to investigate how to adapt them to the task of text classification.

References

Muhammad Abbas, K Ali Memon, A Aleem Jamali, Saleemullah Memon, and Anees Ahmed. 2019. Multinomial naive bayes classification model for sentiment analysis. *IJCSNS Int. J. Comput. Sci. Netw. Secur*, 19(3):62.

Wajdi Aljedaani, Furqan Rustam, Mohamed Wiem Mkaouer, Abdullatif Ghallab, Vaibhav Rupapara, Patrick Bernard Washington, Ernesto Lee, and Imran Ashraf. 2022. Sentiment analysis on twitter data integrating textblob and deep learning models: The case of us airline industry. *Knowledge-Based Systems*, 255:109780.

Hadeel N Alshaer, Mohammed A Otair, Laith Abualigah, Mohammad Alshinwan, and Ahmad M Khasawneh. 2021. Feature selection method using improved chi square on arabic text classifiers: analysis and application. *Multimedia Tools and Applications*, 80:10373–10390.

Chris Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Greg Hullender. 2005. Learning to rank using gradient descent. In *Proceedings of the 22nd International Conference on Machine Learning*, ICML '05, page 89–96, New York, NY, USA. Association for Computing Machinery.

Xinying Chen, Peimin Cong, and Shuo Lv. 2022. A long-text classification method of chinese news based on bert and cnn. *IEEE Access*, 10:34046–34057.

Zhaoqiang Chen, Qun Chen, Boyi Hou, Murtadha Ahmed, and Zhanhuai Li. 2018. Improving machine-based entity resolution with limited human effort: A risk perspective. In *Proceedings of the International Workshop on Real-Time Business Intelligence and Analytics*, BIRTE'18. ACM.

Zhaoqiang Chen, Qun Chen, Boyi Hou, Zhanhuai Li, and Guoliang Li. 2020. Towards interpretable and learnable risk analysis for entity resolution. In *Proceedings of the 2020 ACM SIGMOD international conference on Management of data*, pages 1165–1180.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Abolfazl Farahani, Sahar Voghoei, Khaled Rasheed, and Hamid R Arabnia. 2021. A brief review of domain adaptation. *Advances in data science and information engineering: proceedings from ICDATA 2020 and IKE 2020*, pages 877–894.

Siddhant Garg and Goutham Ramakrishnan. 2020. BAE: BERT-based adversarial examples for text classification. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6174–6181, Online. Association for Computational Linguistics.

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

Dan Hendrycks, Mantas Mazeika, and Thomas Dietterich. 2019. Deep anomaly detection with outlier exposure. In *Proceedings of the 7th International Conference on Learning Representations (ICLR)*, page 1–18.

Heinrich Jiang, Been Kim, Melody Guan, and Maya Gupta. 2018. To trust or not to trust a classifier. In *Advances in Neural Information Processing Systems*, volume 31, page 5541–5552. Curran Associates, Inc.

Shengyi Jiang, Guansong Pang, Meiling Wu, and Limin Kuang. 2012. An improved k-nearest-neighbor algorithm for text categorization. *Expert Systems with Applications*, 39(1):1503–1509.

Zhiying Jiang, Bo Gao, Yanlin He, Yongming Han, Paul Doyle, and Qunxiong Zhu. 2021. Text classification using novel term weighting scheme-based improved tf-idf for internet media reports. *Mathematical Problems in Engineering*, 2021:1–30.

Zhiying Jiang, Matthew Yang, Mikhail Tsirlin, Raphael Tang, Yiqin Dai, and Jimmy Lin. 2023. “low-resource” text classification: A parameter-free classification method with compressors. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 6810–6828, Toronto, Canada. Association for Computational Linguistics.

Rashid Khan, Furqan Rustam, Khadija Kanwal, Arif Mehmood, and Gyu Sang Choi. 2021. Us based covid-19 tweets sentiment analysis using textblob and supervised machine learning algorithms. In *2021 international conference on artificial intelligence (ICAI)*, pages 1–8. IEEE.

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.

Sushil Kumar, VB Singh, and SK Muttoo. 2021. Bug report classification by selecting relevant features using chi square, information gain and latent semantic analysis. In *2021 9th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions)(ICRITO)*, pages 1–5. IEEE.

Jaejun Lee, Raphael Tang, and Jimmy Lin. 2019. What would elsa do? freezing layers during transformer fine-tuning. *arXiv preprint arXiv:1911.03090*.

Haoliang Li, Sinno Jialin Pan, Shiqi Wang, and Alex C Kot. 2018. Domain generalization with adversarial feature learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5400–5409.

Yuxiao Lin, Yuxian Meng, Xiaofei Sun, Qinghong Han, Kun Kuang, Jiwei Li, and Fei Wu. 2021. BertGCN: Transductive text classification by combining GNN and BERT. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1456–1462, Online. Association for Computational Linguistics.

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. *CoRR*, abs/1907.11692.

Yuhan Liu, Saurabh Agarwal, and Shivaram Venkataraman. 2021. Autofreeze: Automatically freezing model blocks to accelerate fine-tuning. *arXiv preprint arXiv:2102.01386*.

Shervin Minaee, Nal Kalchbrenner, Erik Cambria, Narjes Nikzad, Meysam Chenaghlu, and Jianfeng Gao. 2021. Deep learning–based text classification: A comprehensive review. *ACM Comput. Surv.*, 54(3).

Subhabrata Mukherjee and Ahmed Awadallah. 2020. Uncertainty-aware self-training for few-shot text classification. *Advances in Neural Information Processing Systems*, 33:21199–21212.

Youcef Nafa, Qun Chen, Boyi Hou, and Zhanhuai Li. 2024a. Adaptive deep learning for entity disambiguation via knowledge-based risk analysis. *Expert Systems with Applications*, 238:122342.

Youcef Nafa, Qun Chen, Boyi Hou, and Zhanhuai Li. 2024b. Adaptive deep learning for entity disambiguation via knowledge-based risk analysis. *Expert Systems with Application*, 238.

Lin Pan, Chung-Wei Hang, Avirup Sil, and Saloni Potdar. 2022. Improved text classification via contrastive adversarial training. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(10):11130–11138.

Sinno Jialin Pan and Qiang Yang. 2009. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359.

Raghavendra Pappagari, Piotr Zelasko, Jesús Villalba, Yishay Carmiel, and Najim Dehak. 2019. Hierarchical transformers for long document classification. In *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 838–844.

Yi Sun, Yu Zheng, Chao Hao, and Hangping Qiu. 2022. NSP-BERT: A prompt-based few-shot learner through an original pre-training task —— next sentence prediction. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 3233–3250, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Giuseppe Tardivo. 2002. Value at risk (var): The new benchmark for managing market risk. *Journal of Financial Management & Analysis*, 15(1):16–26.

Bruno Trstenjak, Sasa Mikac, and Dzenana Donko. 2014. Knn with tf-idf based framework for text categorization. *Procedia Engineering*, 69:1356–1364.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32.

Wei Ying, Yu Zhang, Junzhou Huang, and Qiang Yang. 2018. Transfer learning via learning to transfer. In *International conference on machine learning*, pages 5085–5094. PMLR.

Lijun Zhang, Xingyu Lu, Zhaoqiang Chen, Tianwei Liu, Qun Chen, and Zhanhuai Li. 2022. Adaptive deep learning for network intrusion detection by risk analysis. *Neurocomputing*, 493:46–58.

Ruohong Zhang, Yau-Shian Wang, and Yiming Yang. 2024a. Generation-driven contrastive self-training for zero-shot text classification with instruction-following LLM. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 659–673, St. Julian's, Malta. Association for Computational Linguistics.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, page 649–657, Cambridge, MA, USA. MIT Press.

Yazhou Zhang, Mengyao Wang, Chenyu Ren, Qiuchi Li, Prayag Tiwari, Benyou Wang, and Jing Qin. 2024b. Pushing the limit of llm capacity for text classification. *arXiv preprint arXiv:2402.07470*.

Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. 2021. A comprehensive survey on transfer learning. *Proceedings of the IEEE*, 109(1):43–76.