Domain Generalization via Switch Knowledge Distillation for Robust Review Representation

You Zhang[†], **Jin Wang**[†*], **Liang-Chih Yu**[‡*], **Dan Xu**[†] and **Xuejie Zhang**[†]
[†]School of Information Science and Engineering, Yunnan University, Yunnan, P.R. China
[‡]Department of Information Management, Yuan Ze University, Taiwan
Contact: wangjin@ynu.edu.cn, lcyu@saturn.yzu.edu.tw

Abstract

Applying neural models injected with in-domain user and product information to learn review representations of unseen or anonymous users incurs an obvious obstacle in content-based recommender systems. For the generalization of the in-domain classifier, most existing models train an extra plain-text model for the unseen domain. Without incorporating historical user and product information, such a schema makes unseen and anonymous users dissociate from the recommender system. To simultaneously learn the review representation of both existing and unseen users, this study proposed a switch knowledge distillation for domain generalization. A generalization-switch (GSwitch) model was initially applied to inject user and product information by flexibly encoding both domain-invariant and domain-specific features. By turning the status `ON` or `OFF`, the model introduced a switch knowledge distillation to learn a robust review representation that performed well for either existing or anonymous unseen users. The empirical experiments were conducted on IMDB, Yelp-2013, and Yelp-2014 by masking out users in test data as unseen and anonymous users. The comparative results indicate that the proposed method enhances the generalization capability of several existing baseline models. For reproducibility, the code for this paper is available at: `https://github.com/yoyo-yun/DG_RRR`.

1 Introduction

With the proliferation of social media, online shopping, and related activities, users are learning to provide an increasing number of reviews about the products they consume (Palmisano et al., 2008; Gauch et al., 2007). With the deluge of customer reviews available, the sentiment score of each customer review can provide implicit feedback for content-based collaborative filtering (Lu et al., 2015).

To learn review representation, previous studies have sought to incorporate external user and product (UP) information into sentiment analysis, which aims to build a neural model by learning contextual and external UP features to predict rating scores (Tang et al., 2015; Chen et al., 2016; Wu et al., 2018; Zhang et al., 2021c,b). The main idea of these models is to inject UP as external feature vectors into sentiment classifiers. These methods can be broadly categorized into two groups according to their injection strategies, i.e., bias- and matrix-based injections. Bias-based methods render bias terms in classifier parameters, while matrix-based methods render matrix terms. Bias-based methods typically perform weaker than matrix-based methods. However, such a method is hard to optimize and cumbersome (Amplayo, 2019). It regards these models as in-domain (ID) distribution for specific users, which collects historical reviews of users on different products and uses them for recommendations.

Due to privacy preservation concerns, some users prefer to use anonymity to bypass the recommender system, which may understand users' intentions through historical interaction. Unfortunately, applying neural models injected with ID UP information to learn review representations of unseen or anonymous users as out-of-domain (OOD) distributions may degrade the classification performance for reviews. As shown in Figure 1, with the ID UP information, neural models could accurately classify positive and negative review samples. However, the performance drops when they face samples with anonymous UP even though textual information is provided.

We hypothesize that degradation occurs mainly because learning review representation depends heavily on external domain-specific features ($\mathrm{F_s}$),

*Corresponding Authors.

Findings of the Association for Computational Linguistics: ACL 2023, pages 12812–12826
July 9-14, 2023 ©2023 Association for Computational Linguistics

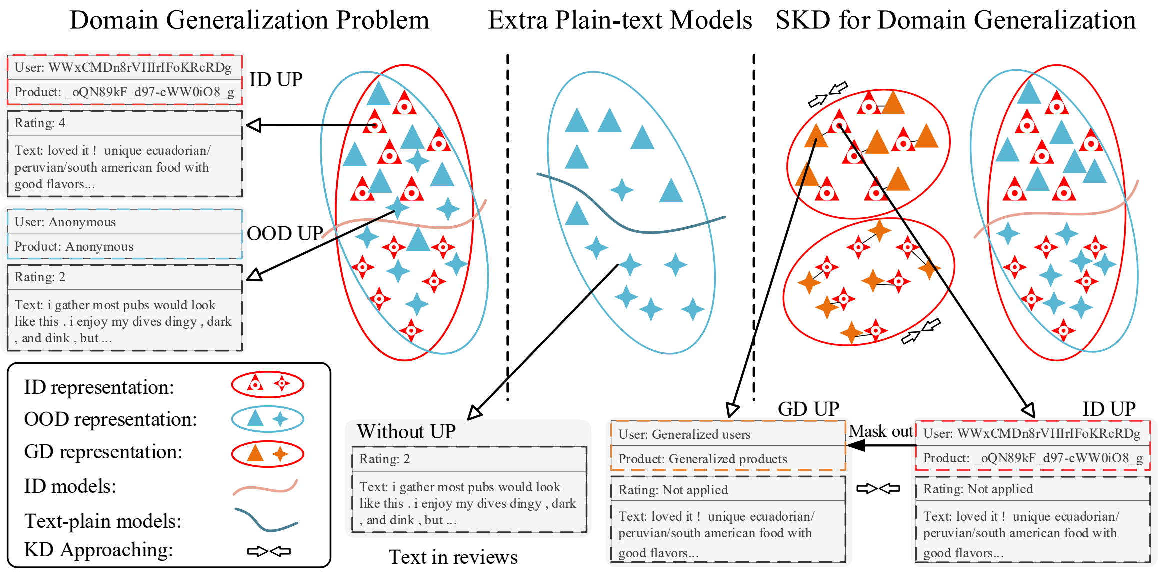

Domain Generalization Problem
Extra Plain-text Models
SKD for Domain Generalization
User: WWxCMDn8rVHIrIFoKRcRDg
Product: _oQN89kF_d97-cWW0iO8_g
ID UP
Rating: 4
Text: loved it ! unique ecuadorian/ peruvian/south american food with good flavors...
User: Anonymous
Product: Anonymous
OOD UP
Rating: 2
Text: i gather most pubs would look like this . i enjoy my dives dingy , dark , and dink , but ...
ID representation:
OOD representation:
GD representation:
ID models:
Text-plain models:
KD Approaching:
Without UP
Rating: 2
Text: i gather most pubs would look like this . i enjoy my dives dingy , dark , and dink , but ...
Text in reviews
GD UP
User: Generalized users
Product: Generalized products
Mask out
Rating: Not applied
Text: loved it ! unique ecuadorian/ peruvian/south american food with good flavors...
ID UP
User: WWxCMDn8rVHIrIFoKRcRDg
Product: _oQN89kF_d97-cWW0iO8_g
Rating: Not applied
Text: loved it ! unique ecuadorian/ peruvian/south american food with good flavors...

Figure 1: Comparisons of generalization performance in review classifications. Neural models injected UP information show high-performing capability of classification for ID UPs while degrading for unseen or anonymous UPs. Applying an extra plain-text model for unseen UPs can generate review representation while dissociating UPs from the recommender system. The proposed method masks out UP information with a generalized UP for sample augmentation. Moreover, a KD strategy facilitate the generalization-domain representation to learn only domain-invariant features from review representation injected specific UPs. Finally, sentiment models can effectively handle with reviews from historical or unseen UPs in inference.

such as UP, while ignoring domain-invariant features ($\mathrm{F_i}$). As a result, the performance of the trained model for unseen or anonymous users is even lower than that of existing models applied to plain texts, i.e., sentiment models using only review text data. A feasible solution is to learn a domain generalization (DG) model using data from UP information as multiple source domains and then distilling $\mathrm{F_i}$ that can be generalized to unseen or anonymous users (Wang et al., 2022; Zhou et al., 2022).

Several strategies have been recently proposed to address DG challenges in wider applications, such as image understanding (Krizhevsky et al., 2017) , speech recognition (Hinton et al., 2012) , and natural language processing (Sarikaya et al., 2014) . The main idea is to eliminate domain shifts and preserve $\mathrm{F_i}$ for robust generalization in the OOD data distribution (Lee et al., 2022) . Nevertheless, the previous studies are not eligible for direct application since the primary change in data distribution occurs in external injected features instead of the review contents.

In this study, we introduce a knowledge distillation (KD) strategy (Gou et al., 2021) with a generalization-switch (GSwitch) module to distill $\mathrm{F_i}$ in review representation for robust generalization. The main idea is to predefine a generalization domain (GD) distribution that preserves $\mathrm{F_i}$ while eliminating $\mathrm{F_s}$ either in ID or OOD distributions for DG. The GSwitch module simulates a GD distribution by initializing the original UP as zeros, which masks out ID information at the input level. Moreover, the GSwitch module provides ON and OFF statuses that easily converts ID or GD representations to each other by turning status. To maximize the mutual information between review representation in ID and GD (Krause et al., 2010; Seo et al., 2022) , a switch KD (SKD) is proposed, where bidirectional Kullback–Leibler (KL) divergence (Kullback and Leibler, 1951) is utilized to measure domain gaps. Due to the removal of ID information in GD review representation, only $\mathrm{F_i}$ is preserved under KD supervision. Review representation in the GD distribution performs more sufficiently in terms of $\mathrm{F_i}$, and better generalization of sentiment models can be leveraged for OOD.

Extensive experiments were conducted on IMDB, Yelp-2013 and Yelp-2014 by masking out UPs in test data for simulating unseen UPs. The results show that the proposed method outperforms several baseline models when anonymous users appear. It does not force learning $\mathrm{F_s}$ for OOD representation and preserves $\mathrm{F_i}$ for unseen or anony-

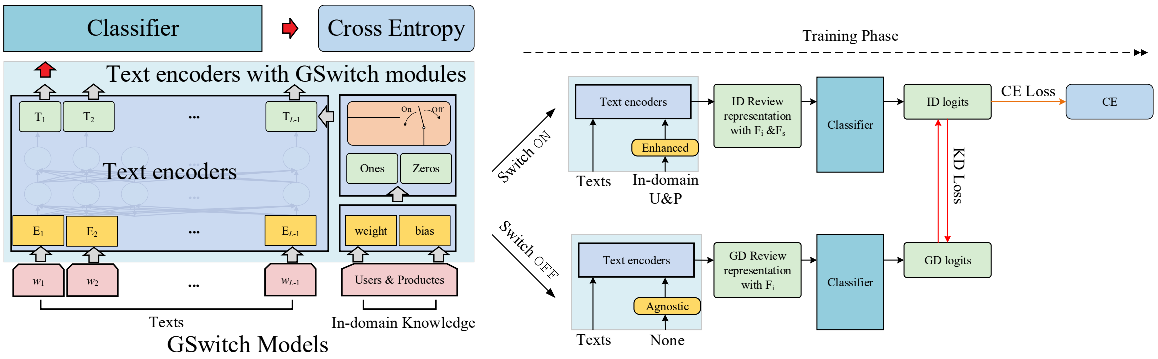


Figure 2: The overview of the proposed method.

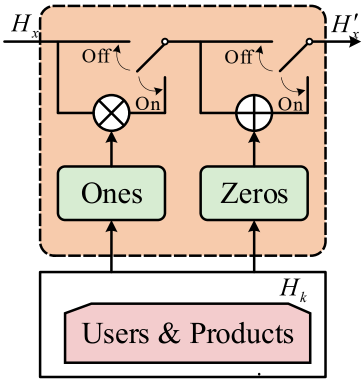


Figure 3: The diagram of GSwitch module.

mous UPs.

The remainder of this paper is organized as follows. A detailed description of the proposed method is illustrated in Section 2. Extensive experiments and analyses are conducted in Section 3. Conclusions are drawn in Sections 4. Finally, the limitations of our paper are discussed in Section 5.

2 Switch Knowledge Distillation

In this section, we elaborate on the GSwitch module and SKD for DG problems in personalized sentiment analysis tasks, as shown in Figure 2. The GSwitch model was proposed to adopt GSwitch modules to inject ID knowledge of UP information into review representation and mask out ID knowledge to simulate GD distribution. SKD is a bidirectional KL divergence-based strategy to enhance GD representation with sufficient $\mathrm{F_i}$ from ID distribution. In practice, unseen or anonymous OOD knowledge in the testing data can also be served with GD representation for DG tasks. Moreover, such OOD knowledge can be further updated for domain-specific knowledge if sufficient domain data are available.

2.1 Domain Generalization

In our DG tasks, there are multiple N source domains with access to the training set $\mathcal{D}^s = \{\mathcal{D}^s_1, \mathcal{D}^s_2, \ldots, \mathcal{D}^s_N\}$ where N denotes the amount of ID knowledge. The ith dataset $\mathcal{D}^s_i = \{(k^{(i)}, x^{(i)}_j, y^{(i)}_j)\}_{j=1}^{M_i} \sim \mathcal{K} \times \mathcal{X} \times \mathcal{Y}$ contains M_i samples associated with a joint distribution $P^{(i)}_{\mathrm{KXY}}$ where $P^{(i)}_{\mathrm{KXY}} \neq P^{(i')}_{\mathrm{KXY}}$ with $i \neq i'$ and $i, i' \in [1 : N]$; where $\mathcal{K}$ and $\mathcal{X}$ denote the input spaces of knowledge and review texts, respectively, and $\mathcal{Y}$ represents the rating space. The aim is to learn a classification function $f(\cdot; \theta) : (\mathcal{K}, \mathcal{X}) \to \mathcal{Y}$ using all source domain data but can also generalize to unseen target domains $\mathcal{D}^t = \{(k^t, x^t)\}$ where $P^t_{\mathrm{KXY}} \neq P^{(i)}_{\mathrm{KXY}}, \forall i \in \{1 : N\}$.

The objective is formulated as follows:

$$\hat{\theta} = \min_{\theta} \mathcal{L} = \mathcal{L}_{\mathrm{CE}} + \beta \mathcal{L}_{\mathrm{MIM}} \tag{1}$$

where $\mathcal{L}_{\mathrm{CE}}$ is the cross-entropy loss applied to each source domain sample and $\mathcal{L}_{\mathrm{MIM}}$ presents the mutual information maximization loss applied to multiple source domains and the corresponding GD distributions with a decay factor β (Krause et al., 2010; Seo et al., 2022). In our work, we introduce a KD loss $\mathcal{L}_{\mathrm{SKD}}$ for instantiating $\mathcal{L}_{\mathrm{MIM}}$; see Sec. 2.3. The learned parameters $\hat{\theta}$ are evaluated on the OOD data in inference.

2.2 Generalization Switch Module

In contrast to the previous DG problem, ID knowledge is explicitly exposed as the input of $k \sim \mathcal{K}$ in our work; therefore, such ID knowledge can be injected into the text representation of $x \sim \mathcal{X}$ to build the joint distribution of P_{KX} via knowledge injection methods in most current models (Zhang et al., 2021b). Conversely, $\mathrm{F_s}$ can also be unloaded from ID models to enhance their generalization performance with the preservation of $\mathrm{F_i}$ (Lu et al., 2022) . To this end, we rethought the previous works and proposed the GSwitch module with two statuses of `ON` and `OFF`.

Given a textual representation of H_x and ID knowledge embeddings of H_k as inputs, the GSwitch module aims to fuse both representations to generate ID textual representation H'_x when its status is `ON`, as shown in Figure 3:

$$H'_x = H_x \odot \mathrm{Linear}_1(H_k) + \mathrm{Linear}_2(H_k) \in \mathbb{R}^{L_x \times d_x} \quad (2)$$

where $H_x \in \mathbb{R}^{L_x \times d_x}$ can be all possible inner textual representations in text encoders, e.g., query, key, and value vectors at multihead attention (MHA) in the transformer structure (Vaswani et al., 2017); for weights and biases, $H_k \in \mathbb{R}^{d_k}$ with the dimensionality of d_k are linearly transformed into d_x consistent with H_x via linear functions of $\mathrm{Linear}_1(\cdot)$ and $\mathrm{Linear}_2(\cdot)$; L_x represents sequence length; and $\odot$ denotes the Hadamard product function (Zhang et al., 2021b,c). The output of the GSwitch module updates original textual representations to knowledge-enhanced ones. When the status is `OFF`, the highway of the GSwitch module deactivates ID knowledge as $H'_x = H_x$. Therefore, the GSwitch module is agnostic to the sentiment model structure.

To further eliminate gaps between the two statuses, we technically set H_k to zero when UPs first participate in the training procedure and reformulate the first term in Eq. (2) as $1 + \mathrm{linear}_1(H_k)$. In such a way, when sentiment models handle unseen UP, the GSwitch module does not affect information propagation either in the training or the inference phase. Such zeroed UPs indicate predefined GD knowledge. This characteristic evolves the GSwitch module as a hot plugin component, especially empowering pretrained language models (PLMs) with well-trained checkpoints. As a result, the GSwitch model (`OFF`) performs as well as the GSwitch model (`ON`) with UPs in GD distributions.

In the training phase, ID models gradually generate ID review representations from the GD distribution as starting points. In inference, ID models encode OOD data as well as GD data since OOD knowledge is zero-initialized as GD knowledge. Therefore, GSwitch models (`ON`) as ID models could generate GD representation by turning their switch status to `OFF`.

Current sentiment classifiers $f(\cdot;\theta)$, such as convolutional NNs (CNNs) (Kim, 2014), long short-term memory (LSTM) (Cheng et al., 2016; Wang et al., 2019), and transform-based PLMs (Qiu et al., 2020), i.e., BERT (Devlin et al., 2019), equipped with the GSwitch module inherits the `ON` and `OFF` status. When the switch status is `ON`, ID review representation is classified with the softmax function, ID logits, $q^T_{\mathrm{ID}} = f(x, k; \theta) \in \mathbb{R}^{d_{rating}}$ is generated from texts and ID knowledge, where d_{rating} is the dimensionality of predicted labels and T is the temperature applied to soften predicted distributions. In contrast, when the switch status is `OFF`, GD logits $q^T_{\mathrm{GD}} = f(x; \theta) \in \mathbb{R}^{d_{rating}}$ are generated from only texts, which is the same as $q^T_{\mathrm{ID}} = f(x, k'; \theta) \in \mathbb{R}^{d_{rating}}$ with the generalization knowledge of zeroed k'.

Algorithm 1 Review Domain Generalization.

```
Input: Sentiment model $\mathcal{M}_{\mathrm{Sentiment}}$ with GSwitch modules $\mathcal{M}_{\mathrm{GSwitch}}$, Source data $\mathcal{D}^{\mathcal{S}}$, confidence threshold $\varepsilon$, and SKD decay factor $\beta$.
Initialization $\mathcal{M}_{\mathrm{Sentiment}}$ randomly initialized or loaded from well-pretrained checkpoints, $k^{\mathcal{S}}$ is zeros.
for iter in $[0 : \mathrm{max\_iter}]$ do
    Sample a batch from source data
    Generate $q_{\mathrm{ID}}$ via $\mathcal{M}_{\mathrm{Sentiment}}$ and $\mathcal{M}_{\mathrm{GSwitch}}$ (ON)
    Generate $q_{\mathrm{GD}}$ via $\mathcal{M}_{\mathrm{Sentiment}}$ and $\mathcal{M}_{\mathrm{GSwitch}}$ (OFF)
    Obtain $\mathcal{L}_{\mathrm{SKD}}$ via Eq. (4).
    Update Model $\mathcal{M}_{\mathrm{Sentiment}}$ and GSwitch modules $\mathcal{M}_{\mathrm{GSwitch}}$ via Eq. (1).
```

2.3 Optimization

Training objective. In the training phase, the cross-entropy loss between q^1_{ID} and y is first applied to multiple source domain data with supervision. To generate $\mathrm{F_i}$, we introduced q^T_{GD} logits for q^T_{IN} eliminating $\mathrm{F_s}$ in review representation. However, q^T_{GD} might not perform well due to a lack of supervised objectives. To address this issue, we introduce a bidirectional KD, namely, SKD, to guide q^T_{GD} to learn $\mathrm{F_i}$ from q^T_{IN}. A detailed procedure is listed in Algorithm 1.

Switch knowledge distillation. To measure the distance of representation between ID and GD distributions, KL divergence is commonly used, formulated as:

$$D_{\mathrm{KL}}(q_{\mathrm{t}} || q_{\mathrm{s}}) = \sum_i q_{\mathrm{t},i} \log \frac{q_{\mathrm{t},i}}{q_{\mathrm{s},i}} \quad (3)$$

where q_{t} presents the teacher probability to guide

Datasets	Ratings	Distributions (%)	Train	Dev	Test	Users	Products
IMDB	1∼10	3/2/3/5/8/13/19/20/12/13	67,426	8,381	9,112	1,310	1,635
Yelp-2013	1∼5	3/8/19/42/27	62,522	7,773	8,671	1,631	1,633
Yelp-2014	1∼5	4/9/20/40/28	183,019	22,745	25,399	4,818	4,194

Table 1: The statistics of the benchmark datasets.

the convergence of the student probability q_{s}; i indexes the ith sample in the source domain data. It is typical to assign q^T_{ID} as the teacher probability since it is empowered with personal knowledge and direct golden-label supervision. We find that it still works with the teacher assignment for q^T_{GD} where the insightful assumption is that applying a penalty term for better generalization as well as improving the relatedness between ID and GD distributions (Ryu et al., 2022; Sun et al., 2022) . As shown in Sec. 3, it is more robust to combine both assumptions. Therefore, the loss of SKD is defined as:

$$\begin{aligned}\mathcal{L}_{\text{SKD}} = \mathbb{E}_{\mathcal{B}_s\sim\mathcal{D}^s}\{\omega\mathbb{E}_{(x,k)\sim F[\mathcal{B}_s;q^1_{\text{ID}}]}D_{\text{KL}}(q^T_{\text{ID}}||q^T_{\text{GD}})\\ +(1-\omega)\mathbb{E}_{(x,k)\sim F[\mathcal{B}_s;q^1_{\text{GD}}]}D_{\text{KL}}(q^T_{\text{GD}}||q^T_{\text{ID}})\}\end{aligned} \tag{4}$$

where $\mathcal{B}_s$ denotes a batch of samples in the source data; $\omega \sim \mathcal{B}(0.5)$ is a random variable sampled from the Bernoulli distribution with a probability of 0.5; and F is a confidence filter to select only reliable predictions to avoid noisy knowledge distillation, defined as:

$$F[\mathcal{D};q] = \{(x,k,y)\in\mathcal{D}|\max(q)>\varepsilon\} \tag{5}$$

where ε is the confidence threshold.

3 Experiments

To investigate the effectiveness of the proposed methods, extensive experiments were conducted on the review sentiment classification task.

3.1 Datasets and Evaluation Metrics

Datasets. We evaluate our method on three personalized sentiment analysis datasets as benchmarks, including IMDB, Yelp-2013, and Yelp-2014 (Tang et al., 2015; Zhang et al., 2021b). All datasets were split into Train $\mathcal{D}_{\text{train}}$, Dev $\mathcal{D}_{\text{dev}}$, and Test $\mathcal{D}_{\text{test}}$. To measure the generalization performance of our method, we chose $\mathcal{D}^s = (x,k,y) \in \mathcal{D}_{\text{train}}$ as the source data with multiple ID knowledge of users and products and define $\mathcal{D}^t = (x,k) \in \mathcal{D}_{\text{test}}$ as testing data where we simulate unseen target domains as all domain knowledge in $\mathcal{D}^t$ is newly participated or anonymous. More details of the datasets are statistically listed in Table 1.

Metrics. Due to all datasets exhibiting unbalanced distribution over ratings, we further adopted Macro-F_1 (F_1) as an additional metric along accuracy (Acc) and rooted mean squared error ($RMSE$) following previous works on personalized sentiment analysis (Tang et al., 2015; Zhang et al., 2021b).

3.2 Implementation Details

Network architecture. Following previous works, we chose well-known neural sentiment classification (NSC) (Cheng et al., 2016) and BERT (Devlin et al., 2019) models as backbone architectures. The attention mechanism (Chaudhari et al., 2021; Yuan et al., 2022) showed high performance in sequence modeling and was selected as a priority injection target in most of the previous state-of-the-art works. Accordingly, we evaluated our injection method, the GSwitch module, primarily on the attention mechanism in NSC and BERT, i.e., hierarchical attention and MHA, namely, GSwitch-NSC (att) and GSwitch-BERT (qkv). Since our method is agnostic to model architecture, more complex injection strategies applied to diverse NNs could be used, as shown in Appendix A.3.

Hyperparameter setting. For GSwitch-BERT, we used the `BERT-base-uncased` version as initial checkpoints in all experiments, available at HuggingFace[1]. With respect to personality knowledge, d_k is set to 256. In terms of SKD, ε and β were selected as 0.3 and 1, respectively. For optimization in GSwitch-BERT, the learning rate was set to 2e-5, the batch size was set to 6 with an acceleration ratio of 4 (virtual 24), and AdamW with a linear schedule was applied. With respect to GSwitch-NSC, the learning rate and batch size were set to 5e-4 and 32, respectively. An early stopping strategy with a patience of 3 epochs was adopted for better generalization and monitoring of the F_1 scores of the Dev set $\mathcal{D}_{\text{dev}}$. All models, in our work, were implemented with PyTorch framework and experiments were conducted on a single RTX 3090 (24G)

[1] `https://huggingface.co/`

Models		IMDB			Yelp-2013			Yelp-2014		
		Acc	$RMSE$	F_1	Acc	$RMSE$	F_1	Acc	$RMSE$	F_1
Plain-text Scenario										
CNN	Ori.	40.5	1.629	-	57.7	0.812	-	58.5	0.808	-
	Rei.	45.61	1.5004	40.42	59.94	0.7465	56.48	60.88	0.7250	56.59
BiLSTM	Ori.	43.3	1.494	-	58.4	0.764	-	59.2	0.733	-
	Rei.	44.21	1.5048	40.92	61.37	0.7117	57.07	62.71	0.6999	59.74
NSC+LA (BiLSTM)	Ori.	48.7	1.381	-	63.1	0.706	-	63.0	0.715	-
	Rei.	48.25	1.3761	44.75	63.30	0.6897	60.89	64.70	0.6642	61.72
BERT	Ori.	51.8	1.191	-	$\underline{67.7}$	$\underline{0.627}$	-	67.2	0.630	-
	Rei.	$\underline{52.21}$	$\underline{1.1633}$	$\underline{49.43}$	67.65	0.6276	$\underline{65.49}$	$\underline{67.65}$	$\underline{0.6153}$	$\underline{65.60}$
Out-of-domain Scenario										
GSwitch-NSC-DG (att)		48.85	1.2828	45.55	64.51	0.6705	62.24	65.31	0.6596	62.95
GSwitch-BERT-DG (qkv)		$\underline{53.05}$	$\underline{1.1414}$	$\underline{50.78}$	$\underline{68.40}$	$\underline{0.6114}$	$\underline{66.82}$	$\underline{68.49}$	$\underline{0.6030}$	$\underline{66.42}$
In-domain Scenario										
NSC+UPA	Ori.	53.3	1.281	-	65.0	0.692	-	66.7	0.654	-
	Rei.	54.31	1.2294	50.42	65.64	0.6683	62.66	68.02	0.6326	64.63
HUAPA		55.0	1.185	-	68.3	0.628	-	68.6	0.626	-
CHIM$_{embedding}$		56.4	1.161	-	67.8	0.646	-	69.2	0.629	-
MA-BERT	Ori.	$\underline{57.3}$	1.042	-	$\underline{70.3}$	$\underline{0.588}$	-	$\underline{71.4}$	$\underline{0.573}$	-
	Rei.	57.28	$\underline{1.0388}$	$\underline{54.02}$	69.87	0.5976	$\underline{67.30}$	71.36	0.5817	$\underline{68.38}$
GSwitch-NSC-ON (att)		54.84	1.1879	50.68	66.23	0.6688	62.67	68.38	0.6327	65.40
GSwitch-BERT-ON (qkv)		$\underline{57.24}$	$\underline{1.0420}$	$\underline{54.45}$	$\underline{70.19}$	$\underline{0.5925}$	$\underline{68.30}$	$\underline{71.14}$	$\underline{0.5848}$	$\underline{68.54}$

Table 2: The comparative test results on plain-text, ID and OOD scenarios. Ori. and Rei. mean the original figures reported in (Zhang et al., 2021b) and the reimplementation according to public available source codes. All figures are averaged over five runs. Underscored figures represent the best performance in each group.

GSwitch-BERT (qkv)	OFF	ON				DG w/SKD			
	$\mathcal{O}$	$\mathcal{I}$	$\mathcal{I} \rightarrow \mathcal{O}$	Avg.	$\nabla\mathcal{O}$	$\mathcal{I}$	$\mathcal{I} \rightarrow \mathcal{O}$	Avg.	$\nabla\mathcal{O}$
IMDB	49.43	54.45	48.17↓	51.31	-1.26↓	53.12↓	50.78↑	51.95↑	+1.35↑
Yelp-2013	65.49	68.30	65.06↓	66.68	-0.43↓	68.76↑	66.82↑	67.79↑	+1.33↑
Yelp-2014	65.60	68.54	65.16↓	66.85	-0.46↓	68.68↑	66.42↑	67.55↑	+0.82↑
GSwitch-NSC (att)	OFF	ON				DG w/SKD			
	$\mathcal{O}$	$\mathcal{I}$	$\mathcal{I} \rightarrow \mathcal{O}$	Avg.	$\nabla\mathcal{O}$	$\mathcal{I}$	$\mathcal{I} \rightarrow \mathcal{O}$	Avg.	$\nabla\mathcal{O}$
IMDB	44.75	50.68	40.61↓	44.75	-4.14↓	51.51↑	45.55↑	48.53↑	+0.80↑
Yelp-2013	60.89	62.67	59.07↓	60.87	-1.82↓	64.12↑	62.24↑	63.18↑	+1.35↑
Yelp-2014	61.72	65.40	60.16↓	62.78	-1.56↓	65.42↑	62.95↑	64.19↑	+1.23↑

Table 3: Meta comparisons of F_1 score in ID and OOD scenarios (denoted as $\mathcal{I}$ and $\mathcal{O}$). $\mathcal{I} \rightarrow \mathcal{O}$ presents models learned in $\mathcal{I}$ and tested to $\mathcal{O}$; Avg. means the average performance between ID and OOD scenarios; $\nabla\mathcal{O}$ denotes the discrepancy between $\mathcal{I} \rightarrow \mathcal{O}$ and $\mathcal{O}$.

GPU device.

Baselines. First, we introduced plain-text models as generalization baselines that were agnostic to ID knowledge in the training and testing phases. These models included CNNs (Kim, 2014), bidirectional LSTM (BiLSTM) (Cheng et al., 2016), NSC based on BiLSTM (Chen et al., 2016), and BERT (Devlin et al., 2019). Next, we also adopted ID models with ID knowledge to evaluate the effectiveness of the proposed GSwitch module for knowledge injection, including NSC+UPA (BiLSTM) (Chen et al., 2016), HUAPA (Wu et al., 2018), CHIM (Amplayo, 2019), and MA-BERT (Zhang et al., 2021b).

3.3 Comparative Results and Discussion

Tables 2 and 3 show the comparative experiments on all three datasets, where Table 2 reports on plain-text, OOD and ID scenarios, and Table 3 focuses on the relatedness among them. Note that plain-text scenarios performed as generalization baselines since domain-specific knowledge was discarded; OOD and ID scenarios meant ID models performed on review data in OOD and ID distributions, respectively. OFF and ON indicate that the GSwitch models were learned from the plain-text and ID scenarios, respectively, and DG indicates that the GSwitch models were learned from the ID scenario via the proposed DG method.

In terms of the plain-text scenario in Table 2, it can be first found that all models achieved comparative results on three metrics. In part, NSC and BERT performed better since the hierarchical structure and pretrained-finetuning learning strategy were introduced, respectively. However, these models might be suboptimal because knowledge-

GSwitch-BERT-GD(kqv)	q_{ID}	q_{GD}	IMDB			Yelp-2013			Yelp-2014		
			$\mathcal{I}$	$\mathcal{I}\to\mathcal{O}$	Avg.	$\mathcal{I}$	$\mathcal{I}\to\mathcal{O}$	Avg.	$\mathcal{I}$	$\mathcal{I}\to\mathcal{O}$	Avg.
$\omega=0$			52.39	49.66	51.03	68.17	66.42	67.30	68.46	66.36	67.41
		✗	52.99	48.11	50.55	68.30	66.23	67.27	68.45	66.07	67.26
$\omega=1$			52.34	49.15	50.74	68.17	66.19	67.18	68.52	66.27	67.40
	✗		52.41	49.02	50.72	67.96	66.23	67.10	68.65	66.23	67.44
$\omega\sim\mathcal{B}(0.5)$	✗	✗	52.97	49.85	51.41	68.57	65.96	67.27	68.66	66.39	67.53
			53.12	**50.78**	**51.95**	**68.76**	**66.82**	**67.79**	**68.68**	**66.42**	**67.55**

Table 4: Comparative F_1 scores of GSwitch-BERT-GD (qkv) with different balances for forward and backward KLs. Detached teacher representation was marked with ✗. **Boldface** figures represent the best performance.

Models	IMDB	Yelp-2013	Yelp-2014
GSwitch-BERT-ON (kqv)	54.45	68.30	68.54
-weight	53.21	67.43	68.47
-bias	52.72	67.05	68.02
GSwitch-NSC-ON (att)	50.68	62.67	65.40
-weight	50.42	62.66	64.63
-bias	50.39	62.59	64.23

Table 5: Ablations of weight- and bias-based injections in GSwitch modules for the ID scenario.

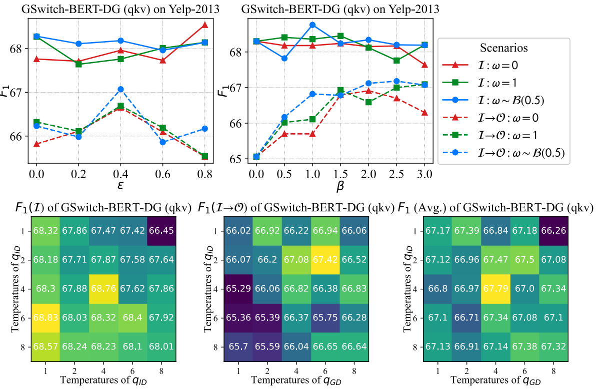


Figure 4: Comparisons with SKD parameters on Yelp-2013. **Upper**: Varying confidence filter threshold ε and decay factor β. **Lower**: Varying different temperatures of domain logits.

potential information was not sufficiently extracted. In a DG method, the proposed models GSwitch-NSC-DG (att) and GSwitch-BERT-DG (qkv) learned performed better in the OOD scenario than plain-text models.

In terms of ID scenarios, we compared the proposed methods with the previous state-of-the-art ID models. Initially, ID models outperformed the first two groups, demonstrating that the introduction of UP information is beneficial to encode reviews. The proposed methods were on par with previous corresponding state-of-the-art models such as GSwitch-BERT-ON (qkv) vs. MA-BERT and GSwitch-NSC-ON (att) vs. NSC+UPA and achieved the best performances in F_1, revealing the effectiveness of the proposed GSwitch modules, where ablation studies can be found in Sec. 3.5.

To further analyze insight DG in personalized sentiment analysis, Table 3 reports clear comparisons with respect to GSwitch-BERT and GSwitch-NSC. From the table, it can be found that although GSwitch models (ON) have achieved better results in comparison with the models (OFF), they failed to be directly applied to OOD scenarios with performance degradation ($\nabla\mathcal{O}$ column). With the introduction of the DG method, such degradation vanished, and the performances on both ID and OOD were leveraged, indicating that $\mathrm{F_i}$ generated from multiple source domain data is beneficial for predictions in OOD data. Unfortunately, GSwitch-BERT (DG) revealed a slight descending trend in $\mathcal{I}$ of IMDB after DG but better performance in a holistic scenario. This finding may be somewhat limited by the relatively complex IMDB task, which covers comprehensive review representations and large-range ratings, leading to overgeneralization applied to ID feature learning. This situation further suggests that caution must be applied in practice.

The findings of both tables can be twofold in sentiment analysis: 1) review representation encoded from only texts can be leveraged via KD adopting ID models as teachers. 2) ID models can be improved by preserving GD review representation with sufficient $\mathrm{F_i}$ for robust generalization performance for unseen UPs.

3.4 Effect of SKD

To evaluate the proposed SKD for DG problems, we conducted several experiments for analysis.

First, to investigate how the combination of bidirectional KDs generates robust representation, we fixed ω to 1 or 0 for comparisons. Table 4 reports the quantitative results. It can be found that either forward or backward KL was feasible to implement knowledge distillation for DG. The forward KL was supported to distill $\mathrm{F_i}$ from ID representations, and backward KL was utilized as a regularization

Models (Yelp-2013)	OFF	ON			GD w/SKD		
	$\mathcal{O}$	$\mathcal{I}$	$\mathcal{I} \to \mathcal{O}$	Avg.	$\mathcal{I}$	$\mathcal{I} \to \mathcal{O}$	Avg.
GSwitch-BERT							
Word embedding	65.49	67.86	65.38	66.62	67.95	65.95	66.95
MHA (qkv)		68.30	65.06	66.68	68.76	66.82	67.79
Feed-forward NNs (ffn)		68.16	65.00	66.58	68.26	66.65	67.46
Document embedding		66.80	65.28	66.04	67.04	66.51	66.78
GSwitch-NSC							
Word embedding	60.89	60.68	54.41	57.55	63.22	61.75	62.49
Hierarchical attention (att)		62.67	59.07	60.87	64.12	62.24	63.18
Document embedding		61.04	56.93	58.99	62.60	60.98	61.79

Table 6: Comparative F_1 performance on Yelp-2013 when different submodels were incorporated with personal knowledge via GSwitch modules.

term applied to ID models. Furthermore, the table presents the performance when back-propagate gradients of teacher logits were detached in KL divergence in Eq. (4). Generally, undetached teacher logits were relatively higher than detached logits. With the combinations between both directions, SKD achieved the best results, indicating its effectiveness.

To further investigate the sensitivity of SKD parameters, Figure 4 illustrates the performance of GSwitch-BERT-DG (qkv) on Yelp-2013 datasets with various crucial parameters, i.e., confidence filter threshold, decay factor, and temperatures. The upper two figures show that when either a larger or lower confidence threshold ε dropped GSwitch-BERT-DG (qkv) performances and appropriate loss decay factors achieved salient balances in both scenarios. The lower figures depict in detail the performances of DG methods in ID and OOD scenarios in inference, as well as their average. With the difference in temperature between q_{ID} and q_{GD}, the performance of the proposed method differed in the ID and OOD scenarios. Accordingly, a larger temperature in q_{ID} than in q_{GD} produced higher performance in the ID scenario and lower performance in the OOD scenario and vice versa. These findings suggest flexible applications according to requirements in practice.

3.5 Effect of GSwitch Modules

GSwitch modules were proposed to be a unified method that rethought the previous works and efficiently model the ID and GD review representation. Table 5 presents an ablation study on weight and bias terms. For the three datasets, GSwitch-BERT-ON (qkv) and GSwitch-NSC-ON (att) achieved the best results. Once either weight or bias terms vanished, the performance dropped concurrently, indicating the effect of GSwitch modules that combined matrix- and bias-based injections.

A clear correlation between injection places and purposes can be surveyed in previous works. In our work, the GSwitch module performed a flexible knowledge injection, and Table 6 presents comparative results with different injection places. It can be found that, with different places to inject, all GSwitch models with the status ON could achieve better results than OFF while failing in OOD scenarios. Meanwhile, the DG method could overcome such failure by building the relatedness between ID and GD distributions, in accordance with previous observations (Sec. 2.3). In particular, when GSwitch modules were injected into submodels with more robust capabilities to model hidden representation, more performance could be leveraged. As seen in Table 6, injection places of MHA, feed-forward NNs, and hierarchical attention revealed higher F_1 scores than other places, consistent with other studies (Chen et al., 2016; Wu et al., 2018; Zhang et al., 2021c).

We also listed the further performances of various models with possible injection places to reveal the flexibility and effectiveness of our works, as shown in Appendix A.3.

4 Conclusions

In this paper, a DG framework with knowledge distillation was proposed to generate robust review representations for sentiment analysis. Rethinking the previous state-of-the-art models, we introduced GSwitch models that connect review representations between ID and GD distributions. To align both representations for sentiment classification, an SKD was proposed, which enables ID models to preserve F_{i} for better generalization on OOD data. Comparative and analytical experiments indicate the effect of GSwitch models and demonstrate that the proposed DG can effectively eliminate domain shifts in sentiment analysis.

5 Limitations

There may be some potential limitations to this work:

- Due to the **maximum input length** limitations and **cumbersome deployments** in most PLMs (i.e., BERT), we limited our **input lengths with a specific selector** (following previous works (Sun et al., 2019; Zhang et al., 2021b)) and **searched hyperparameters in a limited range**, especially in batch sizes (with a maximum batch size of 6). Theoretically, better experimental results can be reported; however, we reimplemented comparative methods and conducted all analytical experiments in the same environments with the same settings, ensuring fairness in performance comparison and problem addressing.
- Due to the characteristics of the applications in our work and the existing DG methods that are difficult to directly apply to our tasks, **we only simulate the performance of plain-text models as DG benchmarks**. However, textual information is inherent in UP-invariant signals for DG performance to some extent, and a comparative experiment indeed leverages the proposed method for better performance in the same OOD scenarios; therefore, it is reasonable for evaluating the performances. To further address these limitations, we will explore more DG strategies to adapt feasible DG methods applied to our personalized sentiment analysis or more complex scenarios with the external introduction of inherent domain shifts in texts such as topics (e.g., books, DVDs, electronics, and kitchen appliances).
- Last but not least, in this paper, **the proposed DG method is only evaluated on personalized sentiment analysis tasks**. However, more applications can be applied to our method, where domain shifts occur due to explicit knowledge injection or F_i can be augmented and exposed.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (NSFC) under Grant Nos. 61966038 and 62266051, the Yunnan Ten Thousand Talents Program and Yunling Scholars Special Project under Grant YNWR-YLXZ-2018-022, the Ministry of Science and Technology (MOST), Taiwan, ROC, under Grant No. MOST110-2628-E-155-002, and the Yunnan Postdoctoral Science Foundation under Grant No. C615300504048. The authors would like to thank the anonymous reviewers for their constructive comments.

References

Reinald Kim Amplayo. 2019. Rethinking Attribute Representation and Injection for Sentiment Classification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP-2019)*, pages 5601–5612.

Reinald Kim Amplayo, Seanie Lee, and Min Song. 2018. Incorporating product description to sentiment topic models for improved aspect-based sentiment analysis. *Information Sciences*, 454-455:200–215.

Yogesh Balaji, Swami Sankaranarayanan, and Rama Chellappa. 2018. MetaReg: Towards Domain Generalization using Meta-Regularization. In *Advances in Neural Information Processing Systems (NeurIPS-2018)*.

Sneha Chaudhari, Varun Mithal, Gungor Polatkan, and Rohan Ramanath. 2021. An Attentive Survey of Attention Models. *ACM Transactions on Intelligent Systems and Technology*, 12(5):1–32.

Huimin Chen, Maosong Sun, Cunchao Tu, Yankai Lin, and Zhiyuan Liu. 2016. Neural Sentiment Classification with User and Product Attention. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP-2016)*, pages 1650–1659.

Jianpeng Cheng, Li Dong, and Mirella Lapata. 2016. Long short-term memory-networks for machine reading. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP-2016)*, pages 551–561.

J. Devlin, M. W. Chang, K. Lee, and K. 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 (NAACL-HLT 2019)*, pages 4171–4186.

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

Susan Gauch, Mirco Speretta, Aravind Chandramouli, and Alessandro Micarelli. 2007. User Profiles for Personalized Information Access. In *The Adaptive Web: Methods and Strategies of Web Personalization*, pages 54–89.

Jianping Gou, Baosheng Yu, Stephen J Maybank, and Dacheng Tao. 2021. Knowledge distillation: A survey. *International Journal of Computer Vision*, 129(6):1789–1819.

Geoffrey Hinton, Li Deng, Dong Yu, George Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara Sainath, and Brian Kingsbury. 2012. Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups. *IEEE Signal Processing Magazine*, 29(6):82–97.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the Knowledge in a Neural Network. *arXiv preprint arXiv:1503.02531*.

Juwon Kang, Sohyun Lee, Namyup Kim, and Suha Kwak. 2022. Style Neophile: Constantly Seeking Novel Styles for Domain Generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR-2022)*, pages 7130–7140.

Sungyeon Kim, Dongwon Kim, Minsu Cho, and Suha Kwak. 2021. Embedding Transfer With Label Relaxation for Improved Metric Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR-2021)*, pages 3967–3976.

Yoon Kim. 2014. Convolutional Neural Networks for Sentence Classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP-2014)*, pages 1746–1751.

Andreas Krause, Pietro Perona, and Ryan Gomes. 2010. Discriminative Clustering by Regularized Information Maximization. In *Advances in Neural Information Processing Systems (NeurIPS-2010)*.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. 2017. ImageNet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90.

Solomon Kullback and Richard A Leibler. 1951. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86.

Kyungmoon Lee, Sungyeon Kim, and Suha Kwak. 2022. Cross-domain Ensemble Distillation for Domain Generalization. In *Computer Vision – European Conference on Computer Vision (ECCV-2022)*, pages 1–20.

Wang Lu, Jindong Wang, Haoliang Li, Yiqiang Chen, and Xing Xie. 2022. Domain-invariant Feature Exploration for Domain Generalization. *Transactions on Machine Learning Research*.

Zhongqi Lu, Zhicheng Dou, Jianxun Lian, Xing Xie, and Qiang Yang. 2015. Content-Based Collaborative Filtering for News Topic Recommendation. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence (AAAI-2015)*, pages 217–223.

C. Palmisano, A. Tuzhilin, and M. Gorgoglione. 2008. Using Context to Improve Predictive Modeling of Customers in Personalization Applications. *IEEE Transactions on Knowledge and Data Engineering*, 20(11):1535–1549.

XiPeng Qiu, TianXiang Sun, YiGe Xu, YunFan Shao, Ning Dai, and XuanJing Huang. 2020. Pre-trained models for natural language processing: A survey. *Science China Technological Sciences*, 63(10):1872–1897.

Minho Ryu, Geonseok Lee, and Kichun Lee. 2022. Knowledge distillation for BERT unsupervised domain adaptation. *Knowledge and Information Systems*, 64(11):3113–3128.

Ruhi Sarikaya, Geoffrey E. Hinton, and Anoop Deoras. 2014. Application of Deep Belief Networks for Natural Language Understanding. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 22(4):778–784.

Seonguk Seo, Joon-Young Lee, and Bohyung Han. 2022. Information-Theoretic Bias Reduction via Causal View of Spurious Correlation. In *Association for the Advancement of Artificial Intelligence (AAAI-2022)*, pages 2180–2188.

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 (CCL-2019)*, pages 194–206.

Tao Sun, Cheng Lu, Tianshuo Zhang, and Haibin Ling. 2022. Safe Self-Refinement for Transformer-Based Domain Adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR-2022)*, pages 7191–7200.

Duyu Tang, Bing Qin, and Ting Liu. 2015. Learning Semantic Representations of Users and Products for Document Level Sentiment Classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (ACL-2015)*, pages 1014–1023.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS-2017)*, pages 5999–6009.

Jin Wang, Liang-chih Yu, K Robert Lai, and Xuejie Zhang. 2019. Tree-Structured Regional CNN-LSTM Model for Dimensional Sentiment Analysis. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*.

Jindong Wang, Cuiling Lan, Chang Liu, Yidong Ouyang, Tao Qin, Wang Lu, Yiqiang Chen, Wenjun Zeng, and Philip Yu. 2022. Generalizing to Unseen Domains: A Survey on Domain Generalization. *IEEE Transactions on Knowledge and Data Engineering*, pages 1–1.

Zhen Wu, Xin Yu Dai, Cunyan Yin, Shujian Huang, and Jiajun Chen. 2018. Improving review representations with user attention and product attention for sentiment classification. In *Proceedings of The Thirty-Second AAAI Conference on Artificial Intelligence (AAAI-18)*, pages 5989–5996.

Ting-Bing Xu and Cheng-Lin Liu. 2019. Data-Distortion Guided Self-Distillation for Deep Neural Networks. *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI-2019)*, pages 5565–5572.

Li Yuan, Jin Wang, Liang-Chih Yu, and Xuejie Zhang. 2022. Syntactic graph attention network for aspect-level sentiment analysis. *IEEE Transactions on Artificial Intelligence*, pages 1–15.

Bo Zhang, Xiaoming Zhang, Yun Liu, Lei Cheng, and Zhoujun Li. 2021a. Matching Distributions between Model and Data: Cross-domain Knowledge Distillation for Unsupervised Domain Adaptation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (ACL-IJCNLP-2021)*, pages 5423–5433.

You Zhang, Jin Wang, Liang-Chih Yu, and Xuejie Zhang. 2021b. MA-BERT: Learning Representation by Incorporating Multi-Attribute Knowledge in Transformers. In *Findings of the Association for Computational Linguistics (ACL-IJCNLP-2021)*, pages 2338–2343.

You Zhang, Jin Wang, and Xuejie Zhang. 2021c. Personalized sentiment classification of customer reviews via an interactive attributes attention model. *Knowledge-Based Systems*, 226:107135.

Kaiyang Zhou, Ziwei Liu, Yu Qiao, Tao Xiang, and Chen Change Loy. 2022. Domain Generalization: A Survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–20.

A Appendix

A.1 Related Work

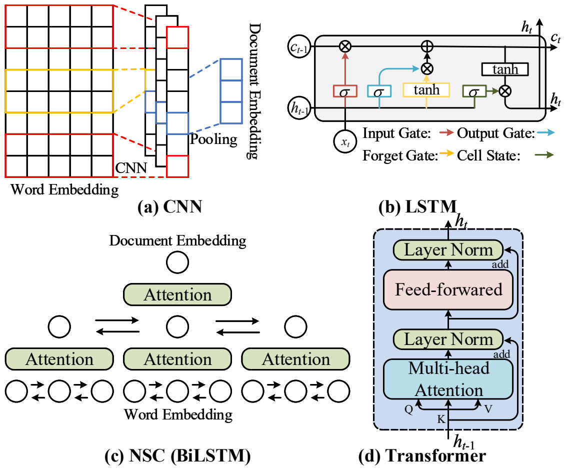
Review sentiment analysis. Review sentiment analysis has attracted increased interest with the utilization of complex review information. Recent studies have proven the effectiveness of the introduction of metaknowledge (i.e., users and products) along with texts via knowledge injection methods. Most of the previous injections can be categorized into matrix- and bias-based methods. Matrix-based methods generally reshape knowledge representations that are initialized or generalized from UP information and then substitute existing submodel weight parameters in text-plain models (Tang et al., 2015; Amplayo, 2019; Zhang et al., 2021b,c). Bias-based methods utilize knowledge representation as biases to be added in text hidden states (Chen et al., 2016; Amplayo et al., 2018; Wu et al., 2018). The final purpose of both categories is to produce UP-specific review hidden states for robust review representation.

In our paper, we combine matrix- and bias-based injections in an efficient way and propose a GSwitch module, which effectively connects to ID and GD review representations to further interact for robust review representation.

Domain generalization. The main aim of DG is to learn a model using data from multiple domains that can then be generalized to unseen domain data (Zhou et al., 2022; Wang et al., 2022). For this purpose, most of the existing approaches mainly contain data augmentation (Kang et al., 2022), $\mathrm{F_i}$ learning (Lu et al., 2022), and meta-learning techniques (Balaji et al., 2018). These models focus on applications such as image understanding (Krizhevsky et al., 2017), speech recognition (Hinton et al., 2012), and natural language processing (Sarikaya et al., 2014).

In contrast to most application scenarios in which $\mathrm{F_s}$ and $\mathrm{F_i}$ are inherently fused as input data, such as cartoon and sketch images with domain-specific styles and domain-invariant contents, in our work, $\mathrm{F_s}$ is mainly injected as external knowledge, and $\mathrm{F_i}$ is primarily located in the text contents themselves. Our work could belong to $\mathrm{F_i}$ learning and data augmentation, where we augment ID review representations to GD ones and then apply the KD strategy to guide GD representations to learn from ID distribution. Since GD representation is agnostic to $\mathrm{F_s}$, only $\mathrm{F_i}$ is preserved.

Knowledge distillation. KD technologies gener-



(a) CNN

(b) LSTM

(c) NSC (BiLSTM)

(d) Transformer

Figure 5: The diagram of four kinds of NNs.

ally transfer knowledge learned from cumbersome teacher models to small student models (Hinton et al., 2015; Gou et al., 2021). It has also been used for other purposes, such as metric learning (Kim et al., 2021), network regularization (Xu and Liu, 2019), and domain adaptation (DA) (Farahani et al., 2021). In particular, for DA, such methods transfer robust knowledge from teacher models learned from source domains to student models applied to target domains. Recently, DA has flexibly introduced distill knowledge methods to further handle specific scenarios where catastrophic forgetting occurs in BERT-based DA (Ryu et al., 2022), and a source-trained model instead of source data is adapted to the target domain for safety in source data (Zhang et al., 2021a). Unfortunately, these KD methods require access to target domain data, while DG tasks serve unseen target domains.

Different from the previous work, we explored KD to transfer $\mathrm{F_i}$ in ID review representations to GD review representations on only source domain data where ID review representation comprises fused information of $\mathrm{F_s}$ and $\mathrm{F_i}$ while GD review representation eliminates $\mathrm{F_s}$.

A.2 Connections to Previous Models

To formulate the injection methods applied to fusion textual and nontextual representation, we mainly survey the previous works in twofold, including bias- and matrix-based methods. First, we formulate encoding procedures of texts and personalities as follows:

$$\begin{aligned} \mathrm{x}' &= f(\mathrm{x}; W_x \in \mathbb{R}^{D_1^x \times D_2^x}, b_x \in \mathbb{R}^{D_2^x}) \\ \mathrm{k}' &= g(\mathrm{k}; W_k \in \mathbb{R}^{D_1^p \times D_2^p}, b_k \in \mathbb{R}^{D_2^p}) \end{aligned} \tag{6}$$

where x and k present text representation and knowledge embeddings, respectively; x′ and k′ present encoded representations; f generally presents a set of language encoding models, i.e., multiple layer perceptron (MLP), CNN, LSTM, and transformer with matrix- and bias-shape parameters (W_x and b_x); g presents a simple linear project or a direct copy ($\mathrm{k}' = \mathrm{k}$) to product knowledge representation. Next, injection methods to generate knowledge-specific representation can be formulated as follows:

$$\mathrm{x}'' = h(\mathrm{x}', \mathrm{k}') = f \circ g = f(\mathrm{x}; W', b') \tag{7}$$

where $\circ$ denotes joint operations, and the injection operation is at the creation of W' and b'.

Bias-based injection. It basically updates the original bias term b_x by:

$$b' = \mathrm{reshape}(\mathrm{k}', D_2^x) + [, b_x] \tag{8}$$

where $\mathrm{reshape}(n, m.shape)$ is a function of reshaping the parameter n with the exact shape of m in a preliminary setting that there is the same number of parameters between n and m.

Based on this assumption, most previous methods (Chen et al., 2016; Wu et al., 2018) take f in Eq. (7-8) as linear projections in self-attention mechanisms to generate knowledge-specific attentive maps for classification.

Matrix-based injection. Another method is to generate knowledge-specific weights via:

$$W' = \mathrm{reshape}(\mathrm{k}', D_1^x \times D_2^x) \tag{9}$$

However, this method might be burdened by many large parameters (Amplayo, 2019), hart to optimize, and with a lack of interactions between textual encoder parameters and knowledge representation (Zhang et al., 2021b). To address these limitations, several works optimize such injections, for instance:

1) CHIM-based method (Amplayo, 2019)

$$\begin{aligned} W' =& W_x \odot \mathrm{repeat}(\hat{W}, C_1 \times C_2) \\ \hat{W} =& \mathrm{reshape}(\mathrm{k}', (D_1^x / C_1) \times (D_2^x / C_2)) \end{aligned} \tag{10}$$

where $\mathrm{repeat}(n, M)$ means repeating parameters n along the corresponding dimensionalities of M.

2) MA-BERT (Zhang et al., 2021b)

$$W' = W_x \odot \mathrm{reshape}(\mathrm{k}', D_2^x) \tag{11}$$

Both CHIM and MA-BERT can efficiently inject knowledge representation into text encoders with further interactions.

Models		OFF	ON						DG w/SKD					
		$\mathcal{O}$	$\mathcal{I}$			$\mathcal{I} \rightarrow \mathcal{O}$			$\mathcal{I}$			$\mathcal{I} \rightarrow \mathcal{O}$		
CNN														
1	Word embedding	40.42	47.18	60.16	61.16	20.54	47.51	36.89	47.04	59.67	60.69	41.20	57.13	57.95
2	Convolutional kernel	56.48	47.22	58.61	60.85	37.12	54.73	54.84	46.61	58.74	60.60	40.58	56.71	57.35
3	Document embedding	56.59	46.78	60.79	60.52	37.93	56.23	56.38	46.51	59.19	60.66	41.26	57.22	57.40
**	Average		47.06	59.85	60.84	31.86	52.82	49.37	46.72	59.20	60.65	41.01	57.02	57.57
BiLSTM														
1	Word embedding		47.47	58.42	60.93	32.40	47.17	43.63	47.68	58.62	61.65	40.53	56.61	58.36
2	Input gate	40.92	47.63	58.96	61.44	38.04	51.96	52.48	48.20	60.28	62.77	41.51	57.32	59.13
3	Output gate		47.78	60.40	62.00	38.35	51.47	57.03	48.74	60.44	62.57	41.32	56.97	59.40
4	Forget gate	57.07	48.22	59.67	62.59	33.52	43.33	52.23	48.42	59.66	62.68	41.78	57.10	60.01
5	Cell memory state		46.57	58.92	62.01	36.60	53.69	56.47	48.24	61.07	62.46	41.42	57.17	59.54
6	Document embedding	59.74	49.11	59.55	61.93	37.66	54.66	57.34	48.04	59.31	62.89	41.53	57.86	60.07
**	Average		47.80	59.32	61.82	36.10	50.38	53.20	48.22	59.90	62.50	41.35	57.17	59.42
NSC (BiLSTM)														
1	Word embedding		50.99	60.68	63.94	35.00	54.41	56.15	51.06	63.22	64.58	43.97	61.75	61.63
2	Sentence embedding	44.75	51.60	60.33	64.06	40.28	55.52	59.58	51.67	60.98	64.31	44.65	59.41	61.16
3	Word-level attention		50.36	63.25	65.07	41.38	59.68	58.70	50.92	63.06	65.10	44.87	61.11	62.54
4	Sentence-level attention	61.72	49.50	61.45	64.10	39.66	59.95	59.12	50.17	62.86	64.51	45.06	61.02	61.93
5	Hierarchical attention		50.68	62.67	65.40	40.61	59.07	60.16	51.51	64.12	65.42	45.55	62.24	62.95
6	Document embedding	60.89	50.85	61.04	63.97	40.88	56.93	58.39	51.39	62.60	64.64	43.24	60.98	62.10
**	Average		50.66	61.57	64.42	39.34	57.59	58.68	51.12	62.81	64.76	44.56	61.09	62.05
Transformer (BERT)														
1	Word embedding		53.89	67.86	68.63	47.47	65.38	65.51	52.78	67.95	68.42	50.01	65.95	66.59
2	MHA (q)		52.94	67.14	67.86	48.24	65.68	64.83	52.06	67.50	68.40	50.37	66.18	66.51
3	MHA (k)		51.74	67.22	67.80	47.52	65.84	65.57	51.12	67.84	68.07	49.83	66.80	66.53
4	MHA (v)	49.43	54.09	67.78	68.38	47.65	65.14	65.01	52.29	68.04	68.33	49.45	65.94	66.23
5	MHA (qkv)		54.45	68.30	68.54	48.17	65.06	65.16	50.78	68.76	68.68	51.95	66.82	66.42
6	Feedforward NN	65.60	53.33	68.16	68.77	46.87	65.00	65.45	52.49	68.26	68.25	48.91	66.65	66.13
7	Layer normalization		53.97	68.05	68.36	48.12	65.31	65.19	52.87	68.39	68.77	50.39	66.83	66.75
8	Residual connection	65.49	53.52	67.98	68.51	47.56	64.64	65.26	52.67	68.36	68.60	50.03	66.59	66.49
9	Document embedding		51.13	66.80	67.53	49.34	65.28	65.23	50.63	67.04	68.06	50.04	66.51	66.98
*	MA-BERT		54.19	67.30	68.38	48.15	64.32	64.90	52.40	67.13	68.35	49.55	65.55	66.45
**	Average		53.33	67.66	68.28	47.91	65.12	65.21	52.01	67.93	68.39	50.05	66.38	66.51

Table 7: Further experiments on diverse injection strategies in four typical NNs. Figures in **orange**, **red**, and **black** represent IMDB, Yelp-2013, and Yelp-2014 datasets, respectively. * represents previous models and ** indicates the average performance of GSwitch modules over different injection places.

In this paper, the proposed method combines Eq. (8) and (11) into PLMs in a more dynamic way via:

$$\mathrm{x}'' = \mathrm{x}' \odot \mathrm{reshape}(\mathrm{k}'_1, D_2^x) + \mathrm{reshape}(\mathrm{k}'_2, D_2^x) \quad (12)$$

which provides a feasible connection between ID knowledge-injected representation and GD representation without knowledge injection and a flexible injection plugged into almost all inner modules of NNs.

A.3 Further Experiments

To further reveal the flexibility of injection and the ability of DG in the proposed method, extensive experimental results are conducted on four kinds of NNs (see Figure 5) for three datasets, as reported in Table 7.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
Section 5

□ A2. Did you discuss any potential risks of your work?
Not applicable. Left blank.

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Section 1.

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☒ Did you use or create scientific artifacts?

Left blank.

□ B1. Did you cite the creators of artifacts you used?
No response.

□ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
No response.

□ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
No response.

□ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
No response.

□ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
No response.

□ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
No response.

C ☑ Did you run computational experiments?

Section 3

□ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
No response.

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Section 3

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Section 3

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Section 3

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

□ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
No response.

□ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
No response.

□ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
No response.

□ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
No response.

□ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
No response.