Towards Domain-Generalizable Paraphrase Identification by Avoiding the Shortcut Learning

Xin Shen
The Chinese University of Hong Kong
xshen@se.cuhk.edu.hk

Wai Lam
The Chinese University of Hong Kong
wlam@se.cuhk.edu.hk

Abstract

In this paper, we investigate the Domain Generalization (DG) problem for supervised Paraphrase Identification (PI). We observe that the performance of existing PI models deteriorates dramatically when tested in an out-of-distribution (OOD) domain. We conjecture that it is caused by shortcut learning, i.e., these models tend to utilize the cue words that are unique for a particular dataset or domain. To alleviate this issue and enhance the DG ability, we propose a PI framework based on Optimal Transport (OT). Our method forces the network to learn the necessary features for all the words in the input, which alleviates the shortcut learning problem. Experimental results show that our method improves the DG ability for the PI models.

1 Introduction

Paraphrase Identification (PI) is the task of recognizing whether one text is a restatement of another text, preserving the same meaning while adopting a different expression (Bhagat and Hovy, 2013). Neural network based models have been proposed for the supervised PI task, and achieve decent performance in the single-domain setting (Yin and Schütze, 2015; Wang et al., 2017; Yang et al., 2019). At present, the existing PI corpora are restricted to several particular domains (Dolan et al., 2004; Xu et al., 2014; He et al., 2020), while the practical sentence pair for the paraphrase judgment can be from any unlabeled domain. At the same time, building a PI corpus for a novel domain needs massive human effort and is expensive. Therefore, a natural question is: for the supervised models trained in the domains with annotated PI corpora, to what extent can they generalize to an out-of-distribution (OOD) domain?

In this paper, we investigate the multi-source (Blanchard et al., 2011) Domain Generalization (DG) (Wang et al., 2021; Zhou et al., 2021) problem for supervised PI. More specifically, we try to learn a PI model based on information from several annotated source domains, and it could generalize well to an unlabeled domain. We investigate several competitive PI models in the DG setting, and observe that their performance deteriorates dramatically when tested in an OOD domain. We conjecture that the poor performance is caused by the models' tendency to the shortcut learning (Geirhos et al., 2020). More specifically, these models are prone to relying on the shortcut features, e.g., some cue words, for classification. These shortcut features are often unique in one particular dataset or domain. When tested on an OOD domain, the models' performance deteriorates because the shortcuts are missing. Interestingly, this phenomenon is also observed in other NLP tasks or models, such as NLI (Gururangan et al., 2018; Du et al., 2021), reading comprehension (Kaushik and Lipton, 2018; Lai et al., 2021), and BERT (Niven and Kao, 2019).

The PI models usually follow the sentence-pair classification paradigm (Lan and Xu, 2018; Devlin et al., 2019). Some models originally proposed for the other sentence pair classification tasks, e.g., Semantic Textual Similarity (STS) or Natural Language Inference (NLI) can be easily adapted to PI. In what follows, we directly apply the suitable models without further clarification. One general character of these models is: they have a component of *information aggregation*, i.e., the extracted and encoded features are aggregated into one fixed-length vector before computing the loss function. We point out that this step is one cause of the shortcut learning problem. Because in this step, it is not uncontrollable that which features should be preserved or discarded in the aggregated vector. Inspired by this, we conduct a new design for the final network layer of PI models to improve the DG

Proceedings of Recent Advances in Natural Language Processing, pages 1318–1325
Sep 1–3, 2021.
https://doi.org/10.26615/978-954-452-072-4_148

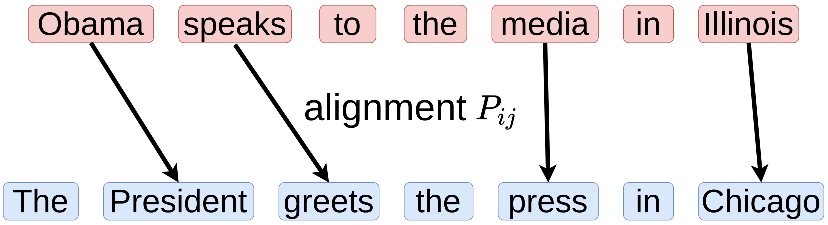

Figure 1: An example of word alignment between two sentences based on OT distance. $\mathbf{P}_{ij}$ is defined in Formula (1). For clarity, the alignment between some unimportant words such as stop words are not shown. This example is also adopted by Kusner et al. (2015).

ability. The motivation of our method is: *if we can force the PI model to learn the necessary features for all the input words instead of just relying on the domain-specific shortcuts, then the effects of shortcut learning can be alleviated.* To this end, our proposed network layer outputs the importance scores and contextualized representations for all the input words, and adopt the Optimal Transport (OT) (Villani, 2008) distance to decide whether two sentences are paraphrase or not. The resulting PI models can be trained end-to-end, the feature extraction and encoding layers are not affected. In the experiments, the PI data from four different domains are adopted for simulating the DG setting. To validate the effectiveness of our method, we consider two representative PI models and equip them with our proposed module. The evaluation results show that our method improve the OOD domain generalizability of these PI models.

2 Problem Formulation

The PI corpus is usually organized as a set $\{((\boldsymbol{x}_i, \widetilde{\boldsymbol{x}}_i), y_i)\}_{i=1}^{N}$. For each tuple $((\boldsymbol{x}, \widetilde{\boldsymbol{x}}), y)$, $\boldsymbol{x}$ and $\widetilde{\boldsymbol{x}}$ are two input sentences, the label $y = 1$ indicates that $\boldsymbol{x}$ and $\widetilde{\boldsymbol{x}}$ are the paraphrase while $y = 0$ denotes the non-paraphrase. The associated *domain* of this dataset is defined as a joint distribution $\mathbb{P}_{\mathcal{X}\mathcal{Y}}$ on $\mathcal{X} \times \mathcal{Y}$, where $\mathcal{X}$ is the space of input sentence pairs[1], and $\mathcal{Y}$ is the label space. Then the target of a PI model is to learn a function $f : \mathcal{X} \rightarrow \mathcal{Y}$, which predicts the label y based on the sentence pair $(\boldsymbol{x}, \widetilde{\boldsymbol{x}})$.

We adopt the common setting of multi-source DG as in Blanchard et al. (2011). Specifically, assume that we can access a set of $K(K > 1)$ distinct source domains $\mathcal{S} = \{\mathcal{S}^k\}_{k=1}^{K}$. Each $\mathcal{S}^k$ is associated with a distinct joint distribution $\mathbb{P}^k_{\mathcal{X}\mathcal{Y}}$, i.e., $\mathbb{P}^k_{\mathcal{X}\mathcal{Y}} \neq \mathbb{P}^{k'}_{\mathcal{X}\mathcal{Y}}, \forall k \neq k'$ and $k, k' \in \{1, \ldots, K\}$. For $\mathcal{S}^k$, the associated dataset contains i.i.d. data $\{((\boldsymbol{x}^k_i, \widetilde{\boldsymbol{x}}^k_i), y^k_i)\}_{i=1}^{N_k}$ sampled from $\mathbb{P}^k_{\mathcal{X}\mathcal{Y}}$. The target domain denoted as $\mathcal{T}$ is associated with a joint distribution $\mathbb{P}^T_{\mathcal{X}\mathcal{Y}}$, where $\mathbb{P}^T_{\mathcal{X}\mathcal{Y}} \neq \mathbb{P}^k_{\mathcal{X}\mathcal{Y}}, \forall k \in \{1, \ldots, K\}$. Then DG problem for PI is defined as: *given the labeled source domains $\mathcal{S}$, we try to learn a model based on information from $\mathcal{S}$ such that the model can generalize well to an unseen domain $\mathcal{T}$.* It should be noted that DG is more challenging than the related settings such as domain adaptation (Patel et al., 2015) or transfer learning (Pan and Yang, 2009). The difference primarily lies in that DG cannot access both the feature distribution and the label distribution of the the target domain $\mathcal{T}$, which makes it more practical for real-world applications.

3 Method Description

3.1 Shortcut Learning Problem

Regardless of the implementation differences, most neural models for the supervised learning of PI follow the sentence pair classification paradigm, i.e., the features from two sentences are extracted and encoded into one vector for the classification (Lan and Xu, 2018). For these approaches, the final output representation fuses features from all the input words together, and we conjecture that it is the reason for the poor OOD generalizability. Concretely, these models are prone to the shortcut learning, i.e., utilizing the features from some cue words that are specific to the training domains. In the fused representations of the final output, the model neglects the features from the words that actually decide whether two sentences are paraphrase or not, because the model already makes the correct decision based on the shortcut features. When these shortcuts are missing in the OOD domain, the model performs poorly.

[1] With a slight abuse of the terminology here, we do not try to rigorously define a *space* containing pairs of sentences.

s1: In only 14 days^{1}, ***US researchers*** have created^{2} an artificial bacteria-eating virus^{3} from synthetic genes^{4}. s2: An artificial bacteria-eating virus^{3} has been made^{2} from synthetic genes^{4} in ***the record time*** of just two weeks^{1}. *Label*: paraphrase; *Domain*: news; *Dataset*: MRPC (Dolan et al., 2004).
s3: how^{1} the optimal solution^{2} to a linear programming problem^{3} changes^{4} as the ***problem data*** are modified. s4: how^{1} changes^{4} in the ***coefficients*** of a linear programming problem^{3} ***affect*** the optimal solution^{2}. *Label*: non-paraphrase; *Domain*: computer science; *Dataset*: PARADE (He et al., 2020).

Table 1: Examples of paraphrase and non-paraphrase text pairs, which come from two different domains. We manually annotate the phrase-to-phrase alignment, and the semantically related phrases are annotated with the same superscript. For the sake of brevity, we do not annotate more detailed word-to-word alignment, and some unimportant words such as stop words are not annotated. We use red italic font to denote the words that cannot be suitably aligned.

3.2 OT Distance for Measuring the Text Similarity

As a preliminary to our method, we introduce the OT distance first. It provides an explainable approach to measuring the text similarity. Concretely, given two pieces of texts $\boldsymbol{x} = [\mathbf{w}_1, \mathbf{w}_2, \cdots, \mathbf{w}_m]$ with m words and $\widetilde{\boldsymbol{x}} = [\widetilde{\mathbf{w}}_1, \widetilde{\mathbf{w}}_2, \cdots, \widetilde{\mathbf{w}}_n]$ with n words, the OT distance between $\boldsymbol{x}$ and $\widetilde{\boldsymbol{x}}$ is defined as:

$$\begin{aligned}\mathcal{D}_{OT}(\boldsymbol{x}, \widetilde{\boldsymbol{x}}) &= \min_{\mathbf{P}\in\prod(\mathbf{a},\mathbf{b})} \langle \mathbf{P}, \mathbf{C} \rangle \\ &= \min_{\mathbf{P}\in\prod(\boldsymbol{\mu},\widetilde{\boldsymbol{\mu}})} \sum_{i=1}^{m} \sum_{j=1}^{n} c(\mathbf{w}_i, \widetilde{\mathbf{w}}_j) \cdot \mathbf{P}_{ij}.\end{aligned} \tag{1}$$

Here, $\mathbf{C}$ stands for the cost matrix, whose element $\mathbf{C}_{ij} = c(\mathbf{w}_i, \widetilde{\mathbf{w}}_j)$ determines the cost of *transporting* the word $\mathbf{w}_i$ to the word $\widetilde{\mathbf{w}}_j$. $c(\mathbf{w}_i, \widetilde{\mathbf{w}}_j)$ is smaller when $\mathbf{w}_i$ and $\widetilde{\mathbf{w}}_j$ are more semantically similar. The matrix $\mathbf{P}$ is the *transport plan*, where $\mathbf{P}_{i,j}$ is larger when $\mathbf{w}_i$ and $\widetilde{\mathbf{w}}_j$ are more closely aligned. $\prod(\boldsymbol{\mu}, \widetilde{\boldsymbol{\mu}}) = \{\mathbf{P} \in \mathbb{R}_{+}^{m\times n} \mid \mathbf{P}\mathbf{1}_n = \boldsymbol{\mu}, \mathbf{P}^T\mathbf{1}_m = \widetilde{\boldsymbol{\mu}}\}$ is the set of all the feasible transport plans. $\langle\cdot,\cdot\rangle$ stands for the Frobenius dot-product between two matrices of the same size. The vectors $\boldsymbol{\mu} = [\mu_1, \cdots, \mu_m]$ and $\widetilde{\boldsymbol{\mu}} = [\widetilde{\mu}_1, \cdots, \widetilde{\mu}_n]$ satisfy that $\sum_{i=1}^{m} \mu_i = \sum_{j=1}^{n} \widetilde{\mu}_j = 1$, and the element μ_i or $\widetilde{\mu}_j$ reflects the relative importance of the corresponding word in the text. In Figure 1, we give an example of OT distance between two sentences. From this example, we can observe that: by solving the optimization problem in Formula (1), the solution matrix $\mathbf{P}$ explicitly aligns the semantically related words. And the optimal objective value of Problem (1) is the distance of moving sentence $\boldsymbol{x}$ to sentence $\widetilde{\boldsymbol{x}}$.

For the values of vectors $\boldsymbol{\mu}$ and $\widetilde{\boldsymbol{\mu}}$, different models behave differently. Word Mover's Distance (WMD) (Kusner et al., 2015) requires that all the words in one text are equally treated, i.e., $\mu_i = \frac{1}{m}(\forall i, 1 \leq i \leq m)$, and $\widetilde{\mu}_j = \frac{1}{n}(\forall j, 1 \leq j \leq n)$. Yokoi et al. (2020) point out that WMD is not suitable for unsupervised STS, because the importance of each word should be differentiated. They propose Word Rotator's Distance (WRD) and adopt the norm of the pretrained embedding as the weight of the corresponding word. For the unsupervised methods WMD and WRD, the values of $\boldsymbol{\mu}$ and $\widetilde{\boldsymbol{\mu}}$ are fixed, and independent of whether the sentence pair $(\boldsymbol{x}, \widetilde{\boldsymbol{x}})$ is paraphrase or not. However, we claim that they are dependent of each other in the supervised setting. Consider some examples in Table 1. Since the nature of paraphrase is semantic equivalence, for the paraphrase sentences (e.g., s1 and s2 in Table 1), the unaligned words (e.g., **US researchers** and **the record time**) are unimportant. Conversely, the key to a non-paraphrase is to find the difference in between. For the non-paraphrase sentences (e.g., s3 and s4 in Table 1), the unaligned words (e.g., **problem data** and **coefficients**) are key to make the difference and are

Algorithm 1 Log-domain Sinkhorn algorithm for computing the entropy-regularized OT distance.

```
Input: $k = 0$, $\boldsymbol{u}^0 = \mathbf{0}_m, \boldsymbol{v}^0 = \mathbf{0}_n$, $K$ is the maximum number of iterations allowed.
while $k < K$ do
    $\boldsymbol{u}^{k+1} = \boldsymbol{u}^k + \varepsilon \log(\boldsymbol{\mu}) - \log\left(\mathbf{R}(\boldsymbol{u}^k, \boldsymbol{v}^k)\mathbf{1}_n\right)$.
    $\boldsymbol{v}^{k+1} = \boldsymbol{v}^k + \varepsilon \log(\widetilde{\boldsymbol{\mu}}) - \log\left(\mathbf{R}(\boldsymbol{u}^{k+1}, \boldsymbol{v}^k)^T \mathbf{1}_m\right)$.
    $k = k + 1$.
end while
Output: $\mathbf{P}^* = \mathbf{R}(\boldsymbol{u}^k, \boldsymbol{v}^k)$.
```

thus important. Another important issue is the value of $c(\mathbf{w}_i, \widetilde{\mathbf{w}}_j)$. For the unsupervised methods such as WMD and WRD, the value of $c(\mathbf{w}_i, \widetilde{\mathbf{w}}_j)$ is fixed. It is usually computed based on the pretrained embeddings of the corresponding words. However, this practice lacks flexibility when representing the word relatedness in the supervised setting. The contextualized word representations should be adopted.

3.3 Domain-Generalizable PI via OT layer

The analysis in Section 3.1 indicates that the shortcut learning problem is caused by the aggregated representation in the classifier layer, which is adopted by most existing PI models. To make the PI models more domain-generalizable, we change the output layer of the network to memorize the necessary features of all the words during the in-domain training. The analysis in Section 3.2 suggests that the values of $\boldsymbol{\mu}$, $\widetilde{\boldsymbol{\mu}}$, and $\mathbf{C}$ should be adaptive in the supervised setting of PI. At the same time, these values are all specific to individual word. Therefore, we parameterize the word importance vectors and the contextualized word embeddings as the learnable outputs of a neural network. The neural network is trained so that the OT distance $\mathcal{D}_{OT}(\boldsymbol{x}, \widetilde{\boldsymbol{x}})$ is minimized for the paraphrase, while is maximized for the non-paraphrase:

$$\begin{cases} \min \ \ \mathcal{D}_{OT}(\boldsymbol{x}, \widetilde{\boldsymbol{x}}) & \text{if } y = 1; \\ \max \ \ \mathcal{D}_{OT}(\boldsymbol{x}, \widetilde{\boldsymbol{x}}) & \text{if } y = 0. \end{cases} \tag{2}$$

In this way, we force the network to memorize representations for each individual word, instead of learning a fused representation. And we expect the shortcut learning problem can be alleviated. For the practical usage, we adopt the following regression based objective:

$$\min_{\theta} \frac{1}{N} \sum_{i=1}^{N} \left\{ \left(\exp\left(-\mathcal{D}_{OT}(\boldsymbol{x}_i, \widetilde{\boldsymbol{x}}_i)\right) - y_i\right)^2 \right\}, \tag{3}$$

where θ denotes the network parameters. The objective in Formula (3) is mathematically equivalent to the objective in Formula (2), but is more numerically stable. Following the practice as in Chen et al. (2019, 2020), we compute the value of cost $c(\mathbf{w}_i, \widetilde{\mathbf{w}}_j)$ as the cosine distance between the corresponding contextualized word representations. We name our method as **D**omain **G**eneralizable **O**ptimal **T**ransport (DG-OT) layer. Except the final output layer, the preceding layers of PI models can be unchanged. In Figure 2, we present the architecture of decomposable attention model (Parikh et al., 2016) when equipped with our proposed DG-OT layer.

3.4 Computing the OT Distance

To incorporate the OT distance into neural networks, we adopt the practice in (Cuturi, 2013; Frogner et al., 2015) and solve the following entropy-regularized OT problem:

$$\min_{\mathbf{P} \in \prod(\boldsymbol{\mu}, \widetilde{\boldsymbol{\mu}})} \langle \mathbf{P}, \mathbf{C} \rangle - \varepsilon H(\mathbf{P}). \tag{4}$$

Here, $H(\mathbf{P})$ is the entropy regularization term defined as: $H(\mathbf{P}) = -\sum_{i,j} \mathbf{P}_{ij}(log(\mathbf{P}_{ij}) - 1)$. ε is a positive hyper-parameter for controlling its relative importance. When the value of ε is small enough, Problem (4) is a good approximation of original OT distance in Formula (1). In this paper, we utilize the Sinkhorn algorithm in the log domain (Chizat et al., 2018; Schmitzer, 2019) to solve Problem (4). The details are presented in Algorithm 1, in which the function $\mathbf{R}(\boldsymbol{u}, \boldsymbol{v})$ is defined as: $\mathbf{R}(\boldsymbol{u}, \boldsymbol{v}) = diag(\exp(\frac{\boldsymbol{u}}{\varepsilon})) \exp(-\frac{\mathbf{C}}{\varepsilon}) diag(\exp(\frac{\boldsymbol{v}}{\varepsilon}))$. After computing the entropy-regularized OT distance between $\boldsymbol{x}_i$ and $\widetilde{\boldsymbol{x}}_i$ with Algorithm 1, we substitute $\mathcal{D}_{OT}(\boldsymbol{x}_i, \widetilde{\boldsymbol{x}}_i)$ in Formula (3) with the objective value of Problem (4). The resulting OT classifier layer is fully differentiable, and the whole PI model can be trained in an end-to-end way.

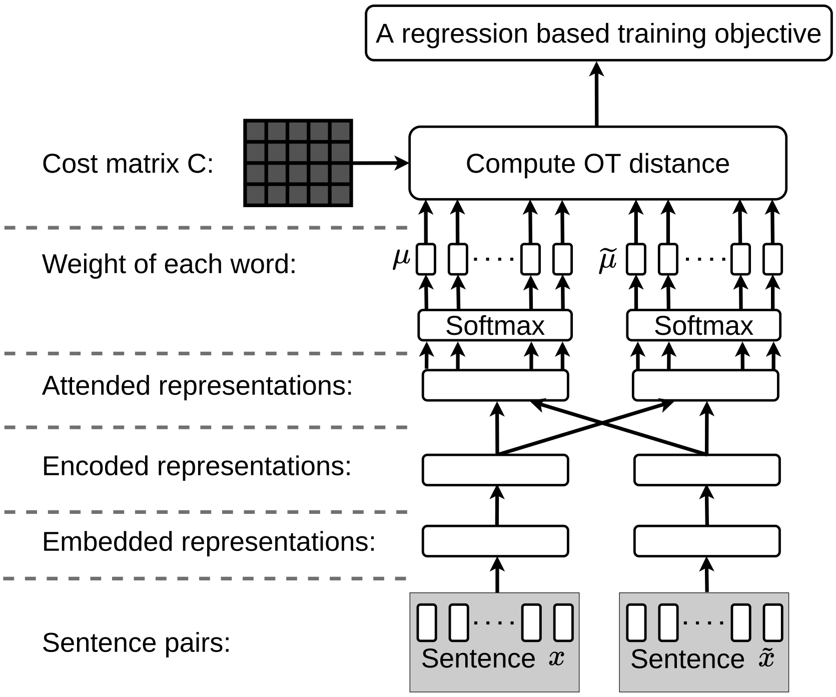

Figure 2: The architecture of decomposable attention model (Parikh et al., 2016) when equipped with our proposed DG-OT layer.

Dataset	**Domain**	**#Training**	**#Validation**	**#Testing**
		total (+/-)	total (+/-)	total (+/-)
MRPC	News	4076 (2753/1323)	500 (250/250)	600 (300/300)
PIT-2015	Twitter	5000 (2500/2500)	1000 (500/500)	1200 (600/600)
QQP	Quora questions	5000 (2500/2500)	1000 (500/500)	1200 (600/600)
PARADE	Computer science	5000 (2500/2500)	1000 (500/500)	1200 (600/600)

Table 2: Statistics of the processed PI datasets. The symbol + indicates the paraphrase sentence pairs, while the symbol - indicates the non-paraphrase sentence pairs.

4 Experiment

4.1 Datasets and Settings

We consider four publicly available PI datasets from different domains for the experiment:

- **M**icrosoft **R**esearch **P**araphrase **C**orpus (**MRPC**) (Dolan et al., 2004), which contains sentence pairs from news articles.
- **P**araphrase **I**dentification from **T**witter (**PIT-2015**) (Xu et al., 2014), which contains pairs of Twitter tweets.
- **Q**uora **Q**uestion **P**airs (**QQP**)[2], which contains Quora question pairs.
- **PARA**phrase identification based on **D**omain knowledg**E** (**PARADE**) (He et al., 2020), which contains definitions of terminologies from the domain of computer science.

To simulate the DG setting, we use three dataset for the in-domain training, and use the remained one dataset for evaluating the OOD generalization ability. During the in-domain training stage, the validation set is merged from three in-domain validation sets. We also conduct the in-domain testing for the purpose of comparison, where the in-domain testing set is merged from three in-domain testing sets. To prevent the PI models from being dominated by one or several particular domains, we process the datasets so that each domain has relatively the same number of sentence pairs. Because the original splittings of these four datasets differ, and it is hard to directly sample training/validation/testing sets and ensure they are of comparative and relatively-large size over different domains. Therefore, for each

[2] https://data.quora.com/First-Quora-Dataset-Release-Question-Pairs.

	source: **NTQ**		*source:* **NTC**		*source:* **NQC**		*source:* **TQC**	
Method	→**NTQ**	→ **C**	→**NTC**	→ **Q**	→ **NQC**	→ **T**	→**TQC**	→ **N**
DecAtt	61.2/63.5	58.3/60.1	56.7/62.9	50.4/58.7	65.5/62.4	56.4/58.1	57.3/63.2	52.9/58.2
DecAtt+DG-OT	63.9/68.8	58.7/64.1	64.8/67.2	62.1/63.3	70.9/68.4	65.4/63.2	63.2/62.8	57.1/60.7
BiMPM	65.2/66.7	62.7/61.4	63.2/68.7	58.3/47.4	68.8/64.9	61.2/53.5	71.8/73.5	52.9/57.7
BiMPM+DG-OT	67.1/70.3	64.1/65.9	63.8/67.5	61.9/57.6	72.1/68.2	65.3/64.9	72.6/73.2	64.5/61.6

Table 3: Results of in-domain testing and OOD generalization. Each result is organized as *accuracy/F1*. We use the initials **N**, **T**, **Q**, **C** to represent the domains of **N**ews, **T**witter, **Q**uora, and **C**omputer science, respectively.

dataset, we merged the original splittings of training/validation/testing sets together, and randomly sample the new training/validation/testing sets. We conduct sampling without replacement. The statistics of the processed datasets are described in Table 2. Following the previous works, we adopt accuracy and F1 score as the evaluation metric.

4.2 Baselines

We adopt the following models for the experiment:

- DecAtt (Parikh et al., 2016), a decomposable attention model. We change the original three-way classification to two-way classification.

- BiMPM (Wang et al., 2017), a bilateral multi-perspective matching model.

For these models, we adopt and adapt the implementations in AllenNLP-Models[3]. To validate the effectiveness of DG-OT layer, we equip these two models with DG-OT layer, and compare them with their vanilla versions. To be fair, the shared network structures have the same size. DecAtt should be compared with DecAtt+DG-OT, and BiMPM should be compared with BiMPM+DG-OT, when the other settings are the same. For all the methods, we adopt GloVe (Global Vectors for Word Representation)[4] to initialize the word embeddings. The hyper-parameters are tuned based on the performance in terms of F1 on the validation set.

4.3 Results

The results of in-domain testing and OOD generalization are reported in Table 3, from which we can draw the following conclusions:

- When other conditions are the same, the OOD performance is poorer than the in-domain performance, which agrees with the common expectation. The reason is the underlying data distributions are different.

- In the same setting, the performance of BiMPM is generally better than that of DecAtt. It suggests that BiMPM is more suitable for the PI task in the setting of multiple-domain training.

- When equipped with DG-OT, both the performance of DecAtt and BiMPM are improved obviously in both the in-domain and OOD setting. And the performance dropping brought by OOD domain is less obvious when DG-OT is equipped by the model. These results validate that DG-OT helps to avoid the shortcut learning.

5 Conclusions and Future Works

As a preliminary attempt in this direction, we investigate the DG problem for supervised PI task in this paper. We point out that the aggregation operation is one reason for the poor OOD generalization ability of the existing PI models. We incorporate the Optimal Transport (OT) distance and design a novel classifier layer, i.e., DG-OT layer. It tackles the shortcut learning problem by enforce the network to learn the importance weights and contextualized representations for all the words. The experiments validate the effectiveness of DG-OT layer. Avoiding the shortcut learning is only one factor to the DG ability of PI models. Another important aspect is how to handle the domain shift, which is left as our future work. Besides, sentence pair classification include other tasks such as STS and NLI. It is still unclear whether our method is suitable for STS and NLI, and we leave this topic as the future work.

[3] https://github.com/allenai/allennlp-models.

[4] *glove.840B.300d* in https://nlp.stanford.edu/projects/glove.

Acknowledgments

The work described in this paper is substantially supported by a grant from the Asian Institute of Supply Chains and Logistics, the Chinese University of Hong Kong.

References

Rahul Bhagat and Eduard Hovy. 2013. Squibs: What is a paraphrase? *Computational Linguistics*, 39(3):463–472.

Gilles Blanchard, Gyemin Lee, and Clayton Scott. 2011. Generalizing from several related classification tasks to a new unlabeled sample. *Advances in neural information processing systems*, 24:2178–2186.

Liqun Chen, Zhe Gan, Yu Cheng, Linjie Li, Lawrence Carin, and Jingjing Liu. 2020. Graph optimal transport for cross-domain alignment. In *Proceedings of the 37th International Conference on Machine Learning*, Proceedings of Machine Learning Research.

Liqun Chen, Yizhe Zhang, Ruiyi Zhang, Chenyang Tao, Zhe Gan, Haichao Zhang, Bai Li, Dinghan Shen, Changyou Chen, and Lawrence Carin. 2019. Improving sequence-to-sequence learning via optimal transport. In *International Conference on Learning Representations*.

Lenaic Chizat, Gabriel Peyré, Bernhard Schmitzer, and François-Xavier Vialard. 2018. Scaling algorithms for unbalanced optimal transport problems. *Mathematics of Computation*, 87(314):2563–2609.

Marco Cuturi. 2013. Sinkhorn distances: Lightspeed computation of optimal transport. In *Advances in Neural Information Processing Systems*, pages 2292–2300.

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.

Bill Dolan, Chris Quirk, and Chris Brockett. 2004. Unsupervised construction of large paraphrase corpora: Exploiting massively parallel news sources. In *COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics*, pages 350–356, Geneva, Switzerland. COLING.

Mengnan Du, Varun Manjunatha, Rajiv Jain, Ruchi Deshpande, Franck Dernoncourt, Jiuxiang Gu, Tong Sun, and Xia Hu. 2021. Towards interpreting and mitigating shortcut learning behavior of nlu models. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 915–929.

Charlie Frogner, Chiyuan Zhang, Hossein Mobahi, Mauricio Araya, and Tomaso A Poggio. 2015. Learning with a wasserstein loss. In *Advances in Neural Information Processing Systems 28*, pages 2053–2061.

Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A Wichmann. 2020. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673.

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel Bowman, and Noah A. Smith. 2018. Annotation artifacts in natural language inference data. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 107–112, New Orleans, Louisiana. Association for Computational Linguistics.

Yun He, Zhuoer Wang, Yin Zhang, Ruihong Huang, and James Caverlee. 2020. PARADE: A New Dataset for Paraphrase Identification Requiring Computer Science Domain Knowledge. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7572–7582, Online. Association for Computational Linguistics.

Divyansh Kaushik and Zachary C. Lipton. 2018. How much reading does reading comprehension require? a critical investigation of popular benchmarks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5010–5015, Brussels, Belgium. Association for Computational Linguistics.

Matt Kusner, Yu Sun, Nicholas Kolkin, and Kilian Weinberger. 2015. From word embeddings to document distances. In *Proceedings of the 32nd International Conference on Machine Learning*, pages 957–966, Lille, France. PMLR.

Yuxuan Lai, Chen Zhang, Yansong Feng, Quzhe Huang, and Dongyan Zhao. 2021. Why machine reading comprehension models learn shortcuts? *Findings of the Association for Computational Linguistics: ACL 2021*.

Wuwei Lan and Wei Xu. 2018. Neural network models for paraphrase identification, semantic textual similarity, natural language inference, and question answering. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3890–3902, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Timothy Niven and Hung-Yu Kao. 2019. Probing neural network comprehension of natural language arguments. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4658–4664, Florence, Italy. Association for Computational Linguistics.

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

Ankur Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. 2016. A decomposable attention model for natural language inference. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2249–2255, Austin, Texas. Association for Computational Linguistics.

Vishal M Patel, Raghuraman Gopalan, Ruonan Li, and Rama Chellappa. 2015. Visual domain adaptation: A survey of recent advances. *IEEE signal processing magazine*, 32(3):53–69.

Bernhard Schmitzer. 2019. Stabilized sparse scaling algorithms for entropy regularized transport problems. *SIAM Journal on Scientific Computing*, 41(3):A1443–A1481.

Cédric Villani. 2008. *Optimal transport: old and new*, volume 338. Springer Science & Business Media.

Jindong Wang, Cuiling Lan, Chang Liu, Yidong Ouyang, and Tao Qin. 2021. Generalizing to unseen domains: A survey on domain generalization. *arXiv preprint arXiv:2103.03097*.

Zhiguo Wang, Wael Hamza, and Radu Florian. 2017. Bilateral multi-perspective matching for natural language sentences. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pages 4144–4150.

Wei Xu, Alan Ritter, Chris Callison-Burch, William B. Dolan, and Yangfeng Ji. 2014. Extracting lexically divergent paraphrases from Twitter. *Transactions of the Association for Computational Linguistics*, 2:435–448.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems*, volume 32, pages 5753–5763. Curran Associates, Inc.

Wenpeng Yin and Hinrich Schütze. 2015. Convolutional neural network for paraphrase identification. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 901–911, Denver, Colorado. Association for Computational Linguistics.

Sho Yokoi, Ryo Takahashi, Reina Akama, Jun Suzuki, and Kentaro Inui. 2020. Word rotator's distance. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2944–2960, Online. Association for Computational Linguistics.

Kaiyang Zhou, Ziwei Liu, Yu Qiao, Tao Xiang, and Chen Change Loy. 2021. Domain generalization: A survey. *arXiv preprint arXiv:2103.02503*.