Distribution Distance Regularized Sequence Representation for Text Matching in Asymmetrical Domains

Weijie Yu, Chen Xu, Jun Xu, *Member, IEEE*, Liang Pang, and Ji-Rong Wen

***Abstract*—Projecting the input text pair into a common semantic space where the matching function can be readily learned is an essential step for asymmetrical text matching. In the practice, it is often observed that the feature vectors from asymmetrical texts show a tendency to be gradually undistinguishable in the semantic space as the model is trained. However, the phenomenon is overlooked in existing studies. As a result, the feature vectors are constructed without any regularization, which inevitably hinders the learning of the downstream matching functions. In this paper, we first exploit the phenomenon and propose DDR-Match, a novel matching framework tailored for asymmetrical text matching. Specifically, in DDR-Match, a distribution distance-based regularizer is devised to accelerate the fusion of sequence representations corresponding to different domains in the semantic space. Then, we provide three instances of DDR-Match and make a comparison among them. DDR-Match is compatible with existing text matching methods by incorporating them as the underlying matching model. Four popular text matching methods are exploited in the paper. Extensive experimental results based on five publicly available benchmarks showed that DDR-Match consistently outperformed its underlying methods.**

***Index Terms*—Text matching, sequence representation, natural language processing.**

I. Introduction

ASYMMETRICAL text matching, which computes the relevance score between textual documents from different domains, is a widely researched area in natural language processing (NLP) and information retrieval (IR). The wide interest of asymmetrical text matching covers a broad spectrum of real-world applications. For example, in natural language inference (NLI), text matching is used to determine whether a hypothesis is an entailment, contradiction, or neutral given a premise [2]. In question answering (QA), text matching is used to determine whether an answer can answer the given question [3], [4]. In IR, text matching is widely used to measure the relevance of a document to a query.

Typical Deep Neural Network-based approaches for asymmetrical text matching are projecting the text sequences from different domains into a common latent space as feature vectors where the matching functions can be readily defined and learned, since these feature vectors have identical dimensions. This type of approach covers a wide range of existing popular matching models, such as DSSM [5], DecAtt [6], RE2 [7], and Sentence-BERT [8]. In real-world matching practices, it is often observed that learning a matching model is a fusion process of the projected sequence vectors in the semantic space. For example, Fig. 1 depicts the distribution of the feature vectors generated by RE2. During the training of RE2 on SciTail dataset [9], it is observed that at the early stage of the training, the feature vectors corresponding to different domains are separately distributed (according to the visualization by tNSE [10]) (Fig. 1(a)). As the model is trained, these separated feature vectors gradually mix together and finally be indistinguishable (Fig. 1(b) and (c)).

The phenomenon can be explained as follows. Given two text sequences from asymmetrical domains (e.g., NLI), the first sequence (e.g., premise) and the second sequence (e.g., hypothesis) are heterogeneous and there exists a lexical gap that needs to be bridged between them [11], similar to that of learning cross-modal matching model. Existing studies [12]–[15] have shown that it is critical for the projection network to generate modal-invariant features. That is, the global distributions of feature vectors should be similar in a common subspace such that their origins cannot be discriminated. This goal is based on the assumption that information concerning modality is noise for tasks requiring only the semantic content of the input. The invariant mappings aim at reducing the modality gap and help capture underlying commonalities and correlated features as aligned projections on the shared subspace. The phenomenon is

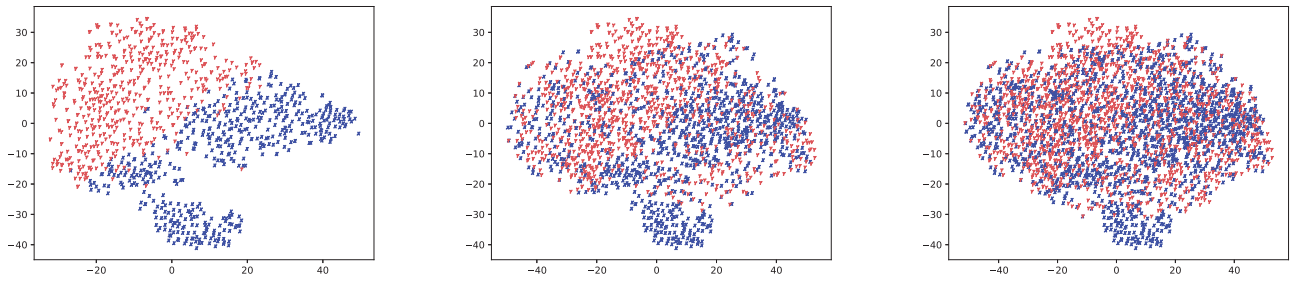
Manuscript received August 7, 2021; revised December 8, 2021; accepted January 14, 2022. Date of publication February 1, 2022; date of current version February 9, 2022. This work was supported in part by the National Key R&D Program of China under Grant 2019YFE0198200, in part by the National Natural Science Foundation of China under Grants 61872338, 61832017, and 62006234, in part by Beijing Outstanding Young Scientist Program under Grant BJJWZYJH012019100020098, and in part by Intelligent Social Governance Interdisciplinary Platform, Major Innovation & Planning Interdisciplinary Platform for the "Double-First Class" Initiative, Renmin University of China, and Public Policy and Decision-making Research Lab of Renmin University of China. This paper is an extension of our EMNLP 2020 paper "Wasserstein Distance Regularized Sequence Representation for Text Matching in Asymmetrical Domains" [1]. The Associate Editor coordinating the review of this manuscript and approving it for publication was Dr. Jianfeng Gao. *(Corresponding author: Jun Xu.)*

Weijie Yu is with the School of Information, Renmin University of China, Beijing 100872, China (e-mail: yuweijie@ruc.edu.cn).

Chen Xu, Jun Xu, and Ji-Rong Wen are with the Beijing Key Laboratory of Big Data Management and Analysis Methods, Gaoling School of Artificial Intelligence, Renmin University of China, Beijing 100872, China (e-mail: xc_chen@ruc.edu.cn; junxu@ruc.edu.cn; jrwen@ruc.edu.cn).

Liang Pang is with the Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China (e-mail: pangliang@ict.ac.cn).

Digital Object Identifier 10.1109/TASLP.2022.3145289



(a) Epoch 1 (b) Epoch 10 (c) Epoch 20

Fig. 1. t-SNE visualization of the projected feature vectors, based on the RE2 models trained on SciTail dataset. Subfigure (a), (b), and (c) respectively illustrates the vector distributions at epochs 1, 10, and 20. The blue 'X' and red 'Y' correspond to the premise and the hypothesis respectively.

not unique but recurs in the experiments based on other matching models and other datasets.

However, existing text matching models ignore this phenomenon and lack constraints or regularizations to ensure that the projected vectors are well distributed for matching. To address this issue, we present a novel learning to match framework, called DDR-Match, to explicitly ensure and accelerate the fusion of the sequence representations corresponding to different domains. Specifically, DDR-Match consists of three components: (1) a feature projection component which jointly projects each pair of text sequences into a latent semantic space, as a pair of feature vectors; (2) a regularizer component which estimates the distance between two distributions corresponding to two sequence representations from asymmetrical domains; (3) a matching component which conducts the matching, also on the same set of projected features. To investigate the effectiveness of the proposed framework, we provide three instances of DDR-Match: the Jensen-Shannon divergence version, the Max Mean Discrepancy version, and the Wasserstein distance version.

The training of DDR-Match amounts to repeatedly interplays between two branches: a regularizer branch that estimates the distribution distance between two sequence representations, and a matching branch that minimizes the distribution distance of sequence representations regularized matching loss. In this way, the minimization of the loss function leads to a learning method not only to minimize the matching loss, but also to well distribute the feature vectors in the semantic space for better matching.

To summarize, this paper makes the following main contributions:

- We highlight the critical importance of the global distribution of the projected feature vectors in matching texts between asymmetrical domains, which has not yet been seriously studied in existing models.
- We propose a new learning to match framework called DDR-Match, in which the text matching model is learned by minimizing distribution distance of sequence representations regularized matching loss. Moreover, we provide three types of instances of DDR-Match and make a comparison among these instances from the theoretical and the practical perspectives.
- We conducted empirical studies on five large-scale benchmarks and incorporated four popular text matching methods into DDR-Match as its underlying models. Experimental results demonstrated that DDR-Match achieved better performance than its underlying models across datasets. Extensive analysis showed the effects of the sequence representations distribution distance-based regularizer in terms of guiding the distributions of feature vectors and improving the matching accuracy.

II. Related Work

In this section, we first review the sequence representation in text matching, then introduce three types of measures of distribution distance including the Jensen–Shannon divergence, the Max Mean Discrepancy, and the Wasserstein distance and their applications.

A. Sequence Representation in Text Matching

Sequence representation lies in the core of text matching. Early works inspired by Siamese architecture assign respective neural networks to encode two input sequences into high-level representations. For example, DSSM [5] is one of the classic representation-based matching approaches to text matching which uses feed-forward neural networks to project a text sequence. To capture the local word-level interaction, CDSSM [16], ARC-I [17] and CNTN [18] change sequence encoder to a convolutional neural network which shares parameters in a fixed size sliding window. To further capture the long-term dependence of a text sequence, a group of recurrent neural network based methods were proposed, including RNN-LSTM [19] and MV-LSTM [20]. Recently, with the help of attention mechanism [6] and graph neural network, the sequence representation is obtained by aligning the sequence itself and the other sequence in the input pairs. For example, CSRAN [21] performs multi-level attention refinement with dense connections among multiple levels. DRCN [22] stacks encoding layers and attention layers, then concatenates all previously aligned results. RE2 [7] introduces a consecutive architecture based on augmented residual connection between convolutional layers and attention layers. These models yield strong performance

on several benchmarks. MUSE [23] represented the interaction between questions and answers on the semantic relations graph. To combine the advantages of aforementioned methods, Enhanced-RCNN [24] first applies BiGRU and CNN to encode sequence, then utilizes soft attention alignment, concatenation, and mean/ max pooling to interact sequence representations, finally exploits the fusion layer to output the prediction. Pre-trained language models, represented by BERT [25], are also widely used in text matching tasks. These models first pre-trained on large corpus, and then can be fine-tuned a specific text matching task in order to learn domain specific knowledge.

B. Measures of Distribution Distance

1) *Jensen–Shannon Divergence (JSD)* [26] is a distribution similarity measurement widely used in natural language processing since is a symmetrized and smoothed version of Kullback-Leibler divergence. JSD For example, JSD has been used to compare social media posts from different social groups [27], [28] or articles from pairs of different years [29], [30], measure letter distribution difference between two display names from an individual across Online Social Networks [31], [32]. However, the Jensen-Shannon divergence is not the true metrics. This metric does not satisfy the triangle inequality. In this way, these methods cannot preserve the transitivity to get effective representations for distributions.

2) *Max Mean Discrepancy (MMD)* [33] aims at minimizing the differences of certain statistics between the source and target distributions. Usually, the statistic can be written as the norm of the difference between distribution feature means in the reproducing kernel Hilbert space (RKHS). MMD has been widely applied in the domain adaptation literature. [34]–[36] extended MMD to comparing distributions in a deep neural network, by introducing an adaptation layer and an additional domain confusion loss to learn representations that is both semantically meaningful and domain invariant. MMD has also been exploited in many other applications. For example, [37] provided a comprehensive evaluation setup by comparing graph statistics such as the degree distribution, clustering coefficient distribution and motif counts for two sets of graphs based on variants of MMD. Tran *et al.* [38] propose a novel application of the Maximum Mean Discrepancy (MMD) approach to information retrieval, which attempts to bridge the gap between the global data distribution and the data distribution for a given individual enterprise.

3) Wasserstein Distance (WD) [39] is a metric based on the theory of optimal transport. It gives a natural measure of the distance between two probability distributions. WD has been successfully used in the Generative Adversarial Networks(GAN) [40] framework. Arjovsky *et al.* [41] propose WGAN which uses the Wasserstein-1 metric as a way to improve the original framework of GAN, to alleviate the vanishing gradient and the mode collapse issues. WD has also been explored to learn the domain-invariant features in domain adaptation tasks. For example, Chen *et al.* [39] propose to minimize the WD between the feature distributions of the source and the target domains, yielding better performance and smoother training than the standard training method with a Gradient Reversal Layer [42]. Tolstikhin *et al.* [43] propose WAE, a generative model that minimizes a penalized form of the WD between the model distribution and the target distribution, for modeling the data distribution. Also, WD has been used in node embedding for graph data, where each node is related with a distribution rather than a real vector [44].

Inspired by their success in variant applications, this paper introduces JSD, MMD, and WD to the asymmetrical text matching, as a regularizer to improve the sequence representations.

III. Our Approach: DDR-Match

In this section, we introduce our proposed matching framework DDR-Match. Briefly, DDR-Match consists of the feature projection component F used to project input sequence pair to the common semantic space; the regularizer component G used to compute the distribution distance between sequence representations; the matching component G used to match the sequence representations.

A. Model Architecture

Suppose that we are given a collection of N instances of sequence-sequence-label triples: $\mathcal{D} = \{(X_i, Y_i, \mathbf{z}_i)\}_{i=1}^{N}$ where $X_i \in \mathcal{X}$, $Y_i \in \mathcal{Y}$, and $\mathbf{z}_i \in \mathcal{Z}$ respectively denote the first sequence, the second sequence, and the label indicating the relationship of X_i and Y_i.

As shown in Fig. 2, DDR-Match consists of three components:

1) The Feature Projection Component: Given a sequence pair (X, Y), it is first processed by the feature projection component F to map sequences to high dimensional feature vectors in the semantic space,

$$[\mathbf{h}^X, \mathbf{h}^Y] = F(X, Y), \tag{1}$$

where the feature projection function F outputs a pair of K-dimensional feature vectors $\mathbf{h}^X$ and $\mathbf{h}^Y$ in the semantic space. We suppose that F is a neural network with a set of parameters $\boldsymbol{\theta}_F$ and all the parameters in $\boldsymbol{\theta}_F$ are sharing for X and Y.

2) The Matching Component: The output vectors from the feature projection component F are then fed to the matching component M in order to output the predicted label $\hat{\mathbf{z}}$

$$\hat{\mathbf{z}} = M([\mathbf{h}^X, \mathbf{h}^Y]), \tag{2}$$

We suppose that M is also a neural network with a set of parameters $\boldsymbol{\theta}_M$.

3) The Regularizer Component: Given two sets of the projected feature vectors $\mathbf{h}^X$ and $\mathbf{h}^Y$, the regularizer component estimates the distribution distance between $\mathbb{P}_F^X$ and $\mathbb{P}_F^Y$, we denote $\mathbb{P}_F^X$ and $\mathbb{P}_F^Y$ are two distributions defined over the two groups of feature vectors $\mathbf{h}^X$ and $\mathbf{h}^Y$ respectively.

$$\begin{aligned}
\mathbb{P}_F^X &\triangleq P\left(\mathbf{h}^X | [\mathbf{h}^X, \mathbf{h}^Y] = F(X, Y) \wedge (X, Y) \sim \mathcal{X} \times \mathcal{Y}\right), \\
\mathbb{P}_F^Y &\triangleq P\left(\mathbf{h}^Y | [\mathbf{h}^X, \mathbf{h}^Y] = F(X, Y) \wedge (X, Y) \sim \mathcal{X} \times \mathcal{Y}\right),
\end{aligned} \tag{3}$$

where '$\sim$' means that the pairs (X, Y) are sampled from the joint space $\mathcal{X} \times \mathcal{Y}$. There are several methods to estimate the

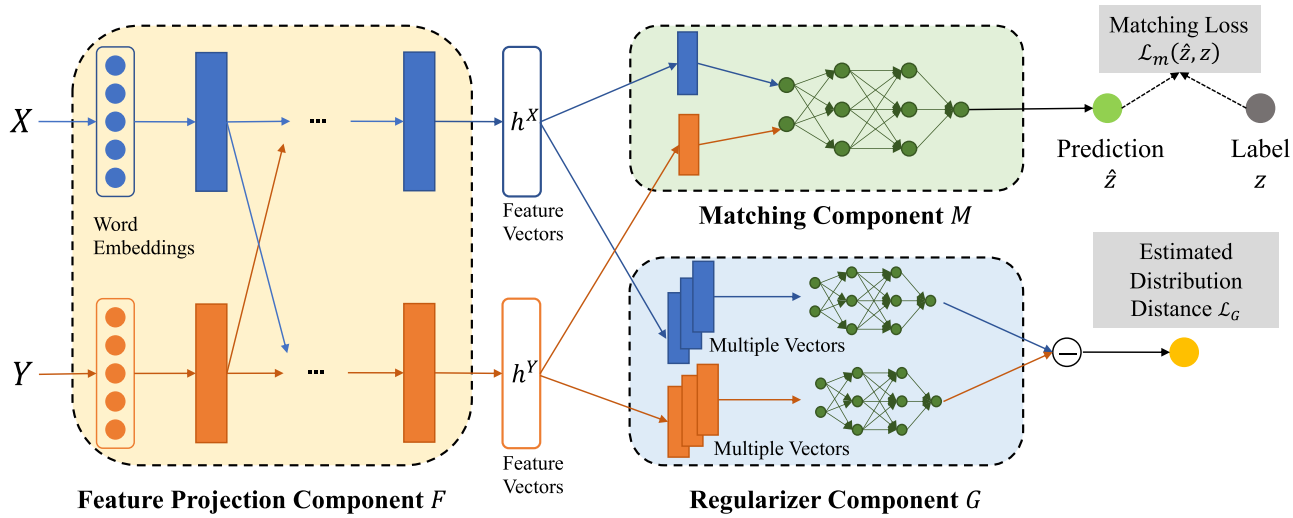


Fig. 2. DDR-Match architecture. F takes text sequences as input and outputs the sequence representations, G takes sequence representations as input and outputs the distance between the sequence representation distributions, M takes sequence representations as input and outputs the label prediction.

distribution distance between $\mathbb{P}_F^X$ and $\mathbb{P}_F^Y$ and different estimations lead to different settings of G. If G is a neural network, we denote the involving parameters in G as $\boldsymbol{\theta}_G$.

B. Optimization

To learn the model parameters $\{\theta_F, \theta_M, \theta_G\}$, DDR-Match sets up two training goals: minimizing the distribution distance between $\mathbb{P}_F^X$ and $\mathbb{P}_F^Y$, and minimizing the prediction loss in terms of the mistakenly predicted matching labels. Therefore, the training process can be divided into two branches: the regularizer branch and the matching branch.

In the regularizer branch, the objective term is based on the estimation of the distribution distance between $\mathbb{P}_F^X$ and $\mathbb{P}_F^Y$. If G is a neural network, the objective term can be written as:

$$\mathcal{O}_G(\theta_G|\theta_F, \theta_M) = \sum_{(X,Y)} G(\mathbf{h}^X, \mathbf{h}^Y), \tag{4}$$

where $[\mathbf{h}^X, \mathbf{h}^Y] = F(X, Y)$ are the projected feature vectors for (X, Y) and G denotes the method which estimates the distance between $\mathbf{h}^X$ and $\mathbf{h}^Y$. Minimizing $\mathcal{O}_G$ w.r.t. the parameters θ_F and θ_G can achieve the goal of reducing the distance between $\mathbb{P}_F^X$ and $\mathbb{P}_F^Y$ in the semantic space. Thus, the regularization loss is defined as:

$$\mathcal{L}_G(\theta_G|\theta_F, \theta_M) = \min \mathcal{O}_G(\theta_G|\theta_F, \theta_M), \tag{5}$$

The matching branch simultaneously updates the matching network M and feature projection network F by seeking the minimization of the sequence representation distribution distance-regularized matching loss:

$$\min_{\boldsymbol{\theta}_F, \boldsymbol{\theta}_M} \mathcal{L}_{reg} = \mathcal{L}_m(\boldsymbol{\theta}_F, \boldsymbol{\theta}_M) + \lambda \cdot \mathcal{L}_G(\theta_G|\theta_F, \theta_M), \tag{6}$$

where $\lambda \in [0, 1]$ is a trade-off coefficient to balance the matching loss and regularizer. The matching loss $\mathcal{L}_m(\boldsymbol{\theta}_F, \boldsymbol{\theta}_M)$ is defined as

$$\mathcal{L}_m(\boldsymbol{\theta}_F, \boldsymbol{\theta}_M) = \sum_{(X,Y,z)\in\mathcal{D}} \ell_m(M(F(X, Y)), \mathbf{z}), \tag{7}$$

where $\ell_m(\cdot, \cdot)$ is the matching loss function defined over each sequence-sequence-label triplet in the training data. It can be, for example, the cross-entropy loss that measures the correctness of the predicted label $\hat{\mathbf{z}} = M(F(X, Y))$ by the matching network, compared to the ground truth label $\mathbf{z}$.

The general procedure of DDR-Match is introduced as follows: DDR-Match takes training set $\mathcal{D} = (X_i, Y_i, z_i)_{i=1}^N$ and a number of hyper-parameters (e.g. the batch size, the learning rate, etc.) as inputs, and outputs the learned parameters θ_F and θ_M. DDR-Match runs multiple rounds until convergence. At each round, DDR-Match maintains two branches. The regularizer branch distribution distance of the projected features $\mathcal{O}_G$ which is constructed based on the sampled sequence pairs (line 4 - line 5). The matching branch estimates the regularized loss $\mathcal{L}_{reg}$ and updates $\boldsymbol{\theta}_F$ and $\boldsymbol{\theta}_M$ (line 8).

DDR-Match is a general framework for asymmetrical text matching. Different configurations of the feature projection component F, the matching component M, and the regularizer component G lead to different matching models. For F and G, existing matching methods are easily incorporated into DDR-Match by means of playing the role as the feature projection component F and the matching component M. For G, any method which is capable of explicitly estimating the distance between distribution pairs can be used as G. As mentioned above, the regularizer component is the key to the DDR-Match, in the next section, we will provide three examples of the regularizer component, achieving three instances of DDR-Match.

IV. Algorithms Derived From DDR-Match

Compared to existing matching methods, the novelty of DDR-Match mainly comes from the regularizer component. In this

Algorithm 1: Training Process for JSD-Based and MMD-Based DDR-Match.

Require: Training set $\mathcal{D} = \{(X_i, Y_i, \mathbf{z}_i)\}_{i=1}^{N}$; mini-batch sizes n; trade-off coefficient λ; learning rates η.

repeat
Sample a mini-batch $\{(X_i, Y_i, \mathbf{z}_i)\}_{i=1}^{n}$ from $\mathcal{D}$
▷ Regularizer branch
if JSD-based regularizer is applied **then**
  $[\mathbf{h}_i^X, \mathbf{h}_i^X] \leftarrow F(X_i, Y_i), \forall i = 1, \ldots, n_1$
  $\mathcal{O}_G(\theta_F) = \sum_{(X,Y)}[\mathbf{h}^X \log(\frac{\mathbf{h}^X}{\mathbf{h}^X+\mathbf{h}^Y}) + \mathbf{h}^Y \log(\frac{\mathbf{h}^Y}{\mathbf{h}^X+\mathbf{h}^Y})]$ {Eq. (9)}
else if MMD-based regularizer is applied **then**
  $[\mathbf{h}_i^X, \mathbf{h}_i^X] \leftarrow F(X_i, Y_i), \forall i = 1, \ldots, n_1$
  $\mathcal{O}_G(\theta_F) = \frac{1}{m^2}\sum_{i,j=1}^{m} k(x_i, x_j) - \frac{2}{mn}\sum_{i,j=1}^{m,n} k(x_i, y_j) + \sum_{i,j=1}^{n} k(y_i, y_j)$ {Eq. (13)}
end if
▷ Matching branch
$\mathcal{L}_{reg} = \sum_{i=1}^{n}[\ell_m(M(F(X,Y)), \mathbf{z}_i) + \lambda\mathcal{O}_G(\theta_F)$
$\{\boldsymbol{\theta}_F, \boldsymbol{\theta}_M\} \leftarrow \{\boldsymbol{\theta}_F, \boldsymbol{\theta}_M\} - \eta_2 \bigtriangledown_{\boldsymbol{\theta}_F, \boldsymbol{\theta}_M} \mathcal{L}_{reg}$ {Eq. (6)}
until convergence
return $\{\boldsymbol{\theta}_F, \boldsymbol{\theta}_M\}$

section, we utilize the JSD, MMD, and WD as the regularizer component of DDR-Match.

A. DDR-Match Model Based on JSD

1) JSD-Based Regularizer: The Jensen–Shannon divergence between two probabilistic distributions $\mathbb{P}_F^X$ and $\mathbb{P}_F^Y$ is defined as:

$$JSD(\mathbb{P}_F^X, \mathbb{P}_F^Y) = \frac{1}{2}KL(\mathbb{P}_F^X||\mathbb{M}) + \frac{1}{2}KL(\mathbb{P}_F^Y||\mathbb{M}) \quad (8)$$

where $\mathbb{M} = \frac{1}{2}(\mathbb{P}_F^X + \mathbb{P}_F^Y)$ and KL denotes the Kullback–Leibler divergence. When applying the Jensen–Shannon divergence to the regularizer component, the objective term is the empirical estimation of JSD:

$$\mathcal{O}_G(\theta_F) = \sum_{(X,Y)}\left[\mathbf{h}^X \log(\frac{\mathbf{h}^X}{\mathbf{h}^X+\mathbf{h}^Y}) + \mathbf{h}^Y \log(\frac{\mathbf{h}^Y}{\mathbf{h}^X+\mathbf{h}^Y})\right] \quad (9)$$

In the JSD-based regularizer, the distance between $\mathbb{P}_F^X$ and $\mathbb{P}_F^Y$ can be directly calculated based on $\mathbf{h}^X$ and $\mathbf{h}^Y$ without any additional parameter, i.e., $\theta_G = \varnothing$. Thus, the loss function for the JSD-based regularizer is:

$$\mathcal{L}_{js}(\boldsymbol{\theta}_F) = \min \mathcal{O}_G(\theta_F). \quad (10)$$

Please note that since there is no learnable parameter for $\mathcal{L}_{js}$, DDR-match based on the JSD regularizer can be trained in an end-to-end fashion. Algorithm 1 provides the general process of DDR-Match based on JSD. Note that $\mathcal{L}_{js}$ still takes $\boldsymbol{\theta}_F$ as parameters because it is calculated on the basis of features generated by F.

B. DDR-Match Model Based on MMD

1) MMD-Based Regularizer: The Maximum Mean Discrepancy between two probabilistic distributions $\mathbb{P}_F^X$ and $\mathbb{P}_F^Y$ is defined as:

$$MMD(\mathbb{P}_F^X, \mathbb{P}_F^Y) = \sup_{f\in\mathcal{F}}(\mathbf{E}_{X\sim\mathbb{P}_F^X}[f(X)] - \mathbf{E}_{Y\sim\mathbb{P}_F^Y}[f(Y)]) \quad (11)$$

where $\mathcal{F}$ is a class of functions $f : \mathcal{X} \to \mathbb{R}, \mathcal{Y} \to \mathbb{R}$. We follow [45] and choose the unit ball in a reproducing kernel Hilbert space (RKHS) $\mathcal{H}$ as the function class f:

$$MMD(\mathbb{P}_F^X, \mathbb{P}_F^Y) = ||\mathbf{E}_{X\sim\mathbb{P}_F^X}[\varphi(X)] - \mathbf{E}_{Y\sim\mathbb{P}_F^Y}[\varphi(Y)]||_{\mathcal{H}}, \quad (12)$$

where $\varphi()$ is a projection function which maps to reproducing kernel Hilbert space. When applying the MMD to the regularizer component, one can use the kernel trick to compute the MMD. The objective term of is the empirical estimation of the squared population MMD:

$$\mathcal{O}_G(\theta_F) = \frac{1}{m^2}\sum_{i,j=1}^{m} k(x_i, x_j) - \frac{2}{mn}\sum_{i,j=1}^{m,n} k(x_i, y_j) + \sum_{i,j=1}^{n} k(y_i, y_j) \quad (13)$$

where $k(\cdot,\cdot)$ is the associated continuous kernel in a universal RKHS $\mathcal{H}$, x_i and x_j are independent random variables with distribution $\mathbb{P}_F^X$, y_i and y_j are independent random variables with distribution $\mathbb{P}_F^Y$. In the MMD-based regularizer, the distance between $\mathbb{P}_F^X$ and $\mathbb{P}_F^Y$ can be calculated with the help of kernel trick so that the only parameter in the component is a hyperparameter, the kernel bandwidth μ, i.e., $\theta_G = \varnothing$. Thus, the loss function for MMD-based regularizer is:

$$\mathcal{L}_{mmd}(\boldsymbol{\theta}_F) = \min \mathcal{O}_G(\theta_F). \quad (14)$$

Please note that since there is no learnable parameter for $\mathcal{L}_{mmd}$, DDR-match based on the MMD regularizer can be trained in an end-to-end fashion. Algorithm 1 provides the general process of DDR-Match based on MMD.

C. DDR-Match Model Based on WD

1) WD-Based Regularizer: The WD between two probabilistic distributions $\mathbb{P}_F^X$ and $\mathbb{P}_F^Y$ is defined as:

$$W(\mathbb{P}_F^X, \mathbb{P}_F^Y) = \inf_{\gamma\in\mathcal{J}(\mathbb{P}_F^X, \mathbb{P}_F^Y)}\int \|X - Y\| d\gamma(X, Y), \quad (15)$$

where $\mathcal{J}(\mathbb{P}_F^X, \mathbb{P}_F^Y)$ denotes all joint distributions, γ stands for (X, Y) that have marginal distributions $\mathbb{P}_F^X$ and $\mathbb{P}_F^Y$. It can be shown that W has the dual form [46]:

$$W(\mathbb{P}_F^X, \mathbb{P}_F^Y) = \sup_{|G|_L\leq 1} E_{\mathbb{P}_F^X}[G(\mathbf{h}^X)] - E_{\mathbb{P}_F^Y}[G(\mathbf{h}^Y)], \quad (16)$$

where '$|G|_L \leq 1$' denotes that the 'sup' is taken over the set of all 1-Lipschitz[1] function G; and function $G : \mathcal{R}^K \to \mathcal{R}$ maps

[1] G is 1-Lipschitz $\Leftrightarrow |G(\mathbf{h}) - G(\mathbf{h}')| \leq |\mathbf{h} - \mathbf{h}'|$ for all $\mathbf{h}$ and $\mathbf{h}'$

TABLE I
STATISTICS OF FOUR DATASET USED IN OUR EXPERIMENT, $|C|$ DENOTES THE NUMBER OF CLASSES AND R DENOTES A RANKING FORMULATION

Dataset	Task	$\|C\|$	Pairs
SNLI	premise-hypothesis	3	570k
SICK	premise-hypothesis	3	9.3k
SciTail	premise-hypothesis	2	27k
TrecQA	question-answer	R	56k
WikiQA	question-answer	R	20k

each K-dimensional feature vector in the semantic space to a real number. When applying the WD to the regularizer component, the objective term is the dual form of WD (Equation (16)), which is approximately written as:

$$\mathcal{O}_G(\theta_G|\theta_F,\theta_M) = -\sum_{(X,Y)} \left[G(\mathbf{h}^X) - G(\mathbf{h}^Y)\right], \qquad (17)$$

Different from the above, in the WD-based regularizer, the distance between $\mathbb{P}_F^X$ and $\mathbb{P}_F^Y$ is approximated by means of a two-layer feed-forward neural network θ_G. Minimizing $\mathcal{O}_G$ w.r.t. the parameters θ_G can be achieved by approximating the WD between $\mathbb{P}_F^X$ and $\mathbb{P}_F^Y$ in the semantic space defined by F:

$$\mathcal{L}_{wd}(\theta_G|\theta_F,\theta_M) = \min_{\boldsymbol{\theta}_G} \mathcal{O}_G(\theta_G|\theta_F,\theta_M). \qquad (18)$$

To make G a Lipschitz function (up to a constant), we follow the practices in [41], all of the parameters in θ_G are always clipped to a fixed range $[-c, c]$, where $c > 0$ is a hyperparameter. In practice, the sequence pairs for training G are randomly sampled from the training set $\mathcal{D}$.

In terms of $\mathcal{L}_{wd}$, Eq. (18) involves the approximation of the WD between $\mathbb{P}_F^X$ and $\mathbb{P}_F^Y$ so that DDR-match in the WD version needs to be updated in an alternative training fashion. Algorithm 2 provides the general process of DDR-Match based on the WD regularizer. Specifically, in the WD version, DDR-Match alternatively maintains two branches. The regularizer branch updates the parameters $\boldsymbol{\theta}_G$, with the $\boldsymbol{\theta}_F$ fixed.[2] It contains a sub-iteration in which the parameters are optimized in an iterative manner: first, objective $\mathcal{O}_G$ is constructed based on the sampled sequence pairs (line 4 - line 6); Then $\boldsymbol{\theta}_G$ is updated with gradient ascent (line 7); Finally, each parameter in $\boldsymbol{\theta}_G$ is clipped to $[-c, c]$ for satisfying the 1-Lipschitz constraint (line 8). The matching branch updates $\boldsymbol{\theta}_F$ and $\boldsymbol{\theta}_M$, with $\boldsymbol{\theta}_G$ fixed. It first samples another mini-batch data from the training data and estimates the regularized loss $\mathcal{L}_{adv}$ using the fixed G (line 11 - line 13). Then, the gradients of the parameters are estimated and used to update the parameters (line 14).

Algorithm 2: Training Process for WD-Based DDR-Match.

Require: Training set $\mathcal{D} = \{(X_i, Y_i, \mathbf{z}_i)\}_{i=1}^N$; mini-batch sizes n_1 and n_2; adversarial training step k; trade-off coefficient λ; learning rates η_1 and η_2; clipping threshold c.

1: **repeat**
2: ▷ Regularizer branch
3: **for** $t = 0$ to k **do**
4: Sample a mini-batch $\{(X_i, Y_i, \mathbf{z}_i)\}_{i=1}^{n_1}$ from $\mathcal{D}$
5: $[\mathbf{h}_i^X, \mathbf{h}_i^X] \leftarrow F(X_i, Y_i), \forall i = 1, \ldots, n_1$
6: $\mathcal{O}_G = \sum_{i=1}^{n_1}[G(\mathbf{h}_i^X) - G(\mathbf{h}_i^Y)]$
7: $\boldsymbol{\theta}_G \leftarrow \boldsymbol{\theta}_G + \eta_1 \bigtriangledown_{\boldsymbol{\theta}_G} \mathcal{O}_G$ {Eq. (18)}
8: ClipWeights$(\theta_G, -c, c)$
9: **end for**
10: ▷ Matching branch
11: Sample a mini-batch $\{(X_i, Y_i, z_i)\}_{i=1}^{n_2}$ from $\mathcal{D}$
12: $[\mathbf{h}_i^X, \mathbf{h}_i^Y] \leftarrow F(X_i, Y_i), \forall i = 1, \ldots, n_2$
13: $\mathcal{L}_{reg} = \sum_{i=1}^{n_2}[\ell_m(M(F(X,Y)), \mathbf{z}_i) + \lambda[G(\mathbf{h}_i^X) - G(\mathbf{h}_i^Y)]]$
14: $\{\boldsymbol{\theta}_F, \boldsymbol{\theta}_M\} \leftarrow \{\boldsymbol{\theta}_F, \boldsymbol{\theta}_M\} - \eta_2 \bigtriangledown_{\boldsymbol{\theta}_F, \boldsymbol{\theta}_M} \mathcal{L}_{reg}$ {Eq. (6)}
15: **until** convergence
16: **return** $\{\boldsymbol{\theta}_F, \boldsymbol{\theta}_M\}$

V. EXPERIMENTS

In this section, we present a comprehensive experimental assessment of DDR-Match.

A. Datasets and Metrics

We use five large-scale publicly matching benchmarks: SNLI (Stanford Natural Langauge Inference) [2], SciTail [9], SICK (Sentences Involving Compositional Knowledge) [47], TrecQA (Text Retrieval Conference Question Answering) [3], and WikiQA (Wikipedia open-domain Question Answering) [4]. Table I provides a summary of the datasets used in our experiments.

SNLI [3] is a benchmark for natural language inference. In SNLI, each data record is a premise-hypothesis-label triple. The premise and hypothesis are two sentences and the label could be "entailment," "neutral," "contradiction," or "-". In our experiments, following the practices in [2], the data with the label "-" are ignored. We follow the original dataset partition. Accuracy is used as the evaluation metric for this dataset.

SICK [4] is used to quantify the degree of semantic relatedness between sentences and the categorizations in terms of the entailment relation between the two sentences (with entailment, contradiction, and neutral as gold labels). SICK consists of about 10,000 English sentence pairs which generated a subset of two existing sets: the ImageFlickr8K dataset [5] and the SemEval-2012 dataset [6]. We follow the original dataset partition. Accuracy is used as the evaluation metric for this dataset.

SciTail [7] is an entailment dataset based on multiple-choice science exams and web sentences. Each record is a premise-hypothesis-label triple. The label is "entailment" or "neutral," because scientific factors cannot contradict. We follow the original dataset partition. Accuracy and F1 score are used as the evaluation metric for this dataset.

[2]Note that the regularizer does not depend on M, given F.

[3]https://nlp.stanford.edu/projects/snli
[4]http://marcobaroni.org/composes/sick.html
[5]https://www.kaggle.com/adityajn105/flickr8k/activity
[6]https://www.cs.york.ac. UK/semeval-2012/
[7]http://data.allenai.org/scitail/

TABLE II
PERFORMANCE COMPARISON ON SNLI, SICK, SCITAIL, WIKIQA, TRECQA TEST SET. DDR-MATCH(·) DENOTES DDR-MATCH WITH THE FIRST PARAMETER AS THE UNDERLYING MODEL, THE SECOND PARAMETER AS THE REGULARIZER. † INDICATES THE STATISTICALLY SIGNIFICANT DIFFERENCE OVER THE UNDERLYING METHOD OF DDR-MATCH WITH $p < 0.05$

	SNLI	SICK	SciTail		WikiQA		TrecQA	
Models	**Acc.**	**Acc.**	**Acc.**	**F1**	**MAP**	**MRR**	**MAP**	**MRR**
RE2	89.00	84.20	86.61	88.73	74.96	76.58	73.78	75.76
DDR-Match (RE2, JS)	88.75	84.30	86.74	88.85	73.93	76.04	74.45†	76.16†
DDR-Match (RE2, MMD)	89.02	84.35	**87.39**†	**89.58**†	74.84	76.24	73.76	75.30
DDR-Match (RE2, WD)	**89.09**†	**85.39**†	87.04†	89.12†	**75.31**†	**76.89**	**74.81**†	**76.30**†
DecAtt	82.50	77.99	81.70	84.61	64.03	65.92	70.62	76.88
DDR-Match (DecAtt, JS)	82.56	78.11	82.07	84.86	64.67†	66.77†	**72.52**†	**77.50**†
DDR-Match (DecAtt, MMD)	**82.78**†	78.11	**83.49**†	**86.23**†	65.04†	66.87†	70.88	76.36
DDR-Match (DecAtt, WD)	82.62†	**79.39**†	82.94†	85.88†	**65.16**†	**67.24**†	72.30†	76.91†
SBERT	83.76	63.31	79.23	83.74	68.38	69.74	68.47	76.57
DDR-Match (SBERT, JS)	84.01	63.35	80.75†	83.68†	68.89†	70.27†	69.44†	76.47†
DDR-Match (SBERT, MMD)	83.91	**65.29**†	81.67†	84.49†	68.85†	70.16†	71.83†	76.93†
DDR-Match (SBERT, WD)	**84.17**†	64.39†	**81.93**†	**84.77**†	**69.04**†	**70.43**†	**72.26**†	**77.26**†
BERT	87.91	86.65	89.56	91.52	77.52	78.95	78.78	80.16
DDR-Match (BERT, JS)	87.90	85.04	90.40†	92.13†	79.03†	80.87†	78.82	80.46
DDR-Match (BERT, MMD)	88.02	86.68	90.22†	91.96†	79.54†	81.21†	**79.62**†	**81.00**†
DDR-Match (BERT, WD)	**88.23**†	**86.72**	**90.53**†	**92.29**†	**79.58**†	**81.23**†	79.11†	80.58†

TrecQA [8] is an answer sentence selection dataset designed for the open-domain question answering setting. We use the raw version TrecQA, questions with no answers or with only positive/negative answers are included. The raw version has 82 questions in the development set and 100 questions in the test set. Mean average precision (MAP) and mean reciprocal rank (MRR) are used as the evaluation metrics for this task.

WikiQA [9] is a retrieval-based question answering dataset based on Wikipedia. We follow the data split of the original paper. This dataset consists of 20.4 k training pairs, 2.7 k development pairs, and 6.2 k testing pairs. We use MAP and MRR as the evaluation metrics for this task.

B. Experimental Setup

To verify the effectiveness of the proposed DDR-Match, we combine our proposed regularizer to representative matching models to form new models under DDR-Match framework. In the experiments, we denote the new models as DDR-Match (A, B). A represents existing matching models, such as RE2 [7], DecATT [6], Sentence-BERT (SBERT) [8] and BERT [25]. B represents our proposed regularizers, such as the JSD-based, MMD-based, and WD-based regularizers.

Specifically, in DDR-Match(RE2), F is stacked blocks which consist of multiple convolution layers and multiple attention layers, and M is an MLP; in DDR-Match(DecAtt), F is an attention layer and an aggregation layer, M is an MLP. Please note that we did not implement the Intra-Sentence Attention in our experiments; in DDR-Match(SBERT), F is a siamese BERT-base[10] model to encode sentence pair separately where two BERT networks have tied weights, and M directly calculates the cosine-similarity between the sentence embedding pair. Please note that in DDR-Match(SBERT), the output embedding of '[CLS]' token represents the sentence embedding. In DDR-Match(BERT), F is a pre-trained BERT-base model, different from DDR-Match(SBERT), F takes the concatenation of the text pair as input. M is an MLP. Please note that we conducted a mean pooling operation to the output of BERT to derive a fixed-size sentence embedding pair then pass to G, and the output of the '[CLS]' token pass to M. The G module for four models are identical: the aforementioned JSD regularizer, MMD regularizer, or WD regularizer. The JSD regularizer is non-parametric; the MMD regularizer depends on the setting of the kernel, for example, if Gaussian kernel is applied, the only parameter of the regularizer is the bandwidth of kernel; the WD regularizer consists of a non-linear projection layer and a linear projection layer.

For all models, the parameters of F and M were directly set as their original settings. In the training, all models were trained using the Adam optimizer with the learning rate η_2 tuned amongst $\{0.0001, 0.0005, 0.001\}$. Batch size n_2 was tuned amongst $\{256, 512, 1024\}$. The trade-off coefficient λ was tuned from [0.0001, 0.01]. The clipping threshold was tuned from [0.1, 0.5]. Word embeddings were initialized with GloVe [48] and fixed during training. For the MMD regularizer, we set the number of Gaussian kernels as 1, and we tuned the kernel width from $\{0.1, 1, 10\}$. The best hyperparameters including early stopping were tuned on the development set.

C. Experimental Results

Table II reports the experimental results of DDR-Match and its underlying models including RE2, DecAtt, SBERT, and BERT. All methods are trained ten times and the average results are reported.

We summarize our observations from Table II as follows: 1) In general, DDR-Match consistently outperformed its underlying methods across five datasets in the case of JSD-based,

[8]https://github.com/castorini/NCE-CNN-Torch/tree/master/ data/TrecQA
[9]https://www.microsoft.com/en-us/download/details.aspx?id=52419
[10]https://github.com/google-research/bert

MMD-based, and WD-based regularizers. The results indicate that DDR-Match is capable of improving the matching accuracy for text matching models in asymmetrical domains benefited from the regularizer component. 2) From the results, we found that DDR-Match under the condition of the WD regularizer consistently outperformed its underlying methods and achieved the best performances in 13 out of 20 sets of experiments. The results indicated the effectiveness of the WD-based regularizer. We supposed the advantages of the WD-based regularizer over the JSD-based and the MMD-based regularizers attributed to the stability of the optimization process [41] and its learnable nature. WD-based regularizer not only provide gradient even if two sequence representation distributions are disjoint [41], but also can be estimated using neural networks rather than depending on hyperparameters. Therefore, in practice, DDR-Match incorporated WD-based regularizer achieves better performance than that of the other two regularizers. We will verify our viewpoint in Section V-E. 3) DDR-Match under the condition of the MMD-based regularizer consistently achieved better performances over its underlying methods. The results indicated the effectiveness of the MMD-based regularizer. We supposed the reason why the MMD-based regularizer suits asymmetrical text matching is that MMD is easily estimated as an empirical mean which is concentrated around the true value of the MMD. MMD-based regularizer is responsible for finding the RKHS function that maximizes the difference in expectations between the two probability distributions of sequence representations. [33], [45]. 4) DDR-Match under the condition of the JSD-based regularizer showed improvement on Scitail and TrecQA but performed poorly on SNLI, SICK, and WikiQA. We suppose that JSD-based regularizer cannot provide effective regularization to the feature vectors when two distributions of sequence representation are disjoint. It is because in this case, the JSD is equal to zero and cannot provide gradients [49]. It is because in this case, JSD is equal to zero and the JSD-based regularizer cannot provide gradients to make sequence representations indistinguishable [41], [49] at the beginning of the training. We will verify our viewpoint in Section V-E and Section V-F.

Summarizing the results above, we conclude that DDR-Match is a general while strong framework that can improve different matching models by using them as its underlying matching model.

D. Complexity of Regularizer Component G

As described in Section III, DDR-Match introduces a regularizer component G to existing sequence matching models. In this section, we discuss the complexity of the regularizer component G.

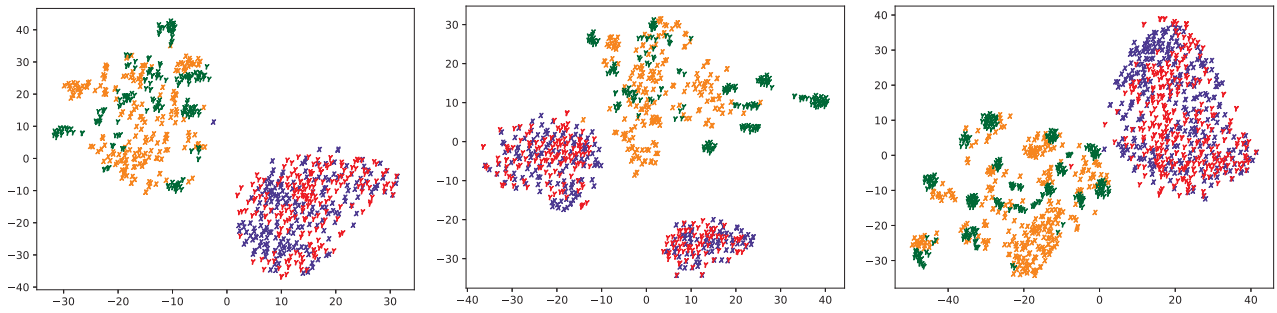
In the case of the JSD-based regularizer, the distance between two sequence representation distributions are calculated according to Eq. (8) and there is no additional parameter. In the case of the MMD-based regularizer, the distance between two sequence representation distributions are calculated according to Eq. (12) and the only additional hyperparameter come the bandwidth of the kernel. In the case of the WD-based regularizer, G module is implemented as a two-layer MLP (the number of neurons in the second layer is set as one). Therefore, the additional computing cost comes from the training of the two-layer MLP, which is of $O(T * N * K * 1)$, where T is the number of training iterations, N number of training examples, K number of neurons in the first layer of MLP (without considering the compute cost of the activation function). We can see that the additional computing overhead is much lower than that of the underlying methods which usually learn much more complex neural networks for the feature projection and the matching. We listed the number of parameters of different text matching models with or without DDR-Match framework in Table III. Compared to the underlying model, the additional parameters of DDR-Match come from the regularizer component G. The results from Table III show that the number of parameters introduced by regularizer component G are far less that of the underlying model.

TABLE III
COMPARISON OF THE NUMBER OF MODEL PARAMETERS AND THE TRAINING TIME (S/BATCH) ON SCITAIL ON A SINGLE NVIDIA TESLA V100 16 GB. THE BATCH SIZE OF RE2-BASED, DECATT-BASED, SBERT-BASED, AND BERT-BASED MODELS ARE SET AS 128, 128, 8, AND 8 RESPECTIVELY. DDR-MATCH(·) DENOTES DDR-MATCH WITH THE FIRST PARAMETER IN PARENTHESES AS THE UNDERLYING MODEL, THE SECOND PARAMETER AS THE REGULARIZER

Models	#Params	Time
RE2	2.8M	0.12
DDR-Match (RE2, JS)	2.8M	0.12
DDR-Match (RE2, MMD)	2.8M	0.13
DDR-Match (RE2, WD)	2.9M	0.15
DecAtt	0.26M	0.06
DDR-Match (DecAtt, JS)	0.26M	0.06
DDR-Match (DecAtt, MMD)	0.26M	0.06
DDR-Match (DecAtt, WD)	0.30M	0.07
SBERT	0.11B	0.25
DDR-Match (SBERT, JS)	0.11B	0.26
DDR-Match (SBERT, MMD)	0.11B	0.28
DDR-Match (SBERT, WD)	0.11B	0.31
BERT	0.11B	0.26
DDR-Match (BERT, JS)	0.11B	0.27
DDR-Match (BERT, MMD)	0.11B	0.28
DDR-Match (BERT, WD)	0.11B	0.32

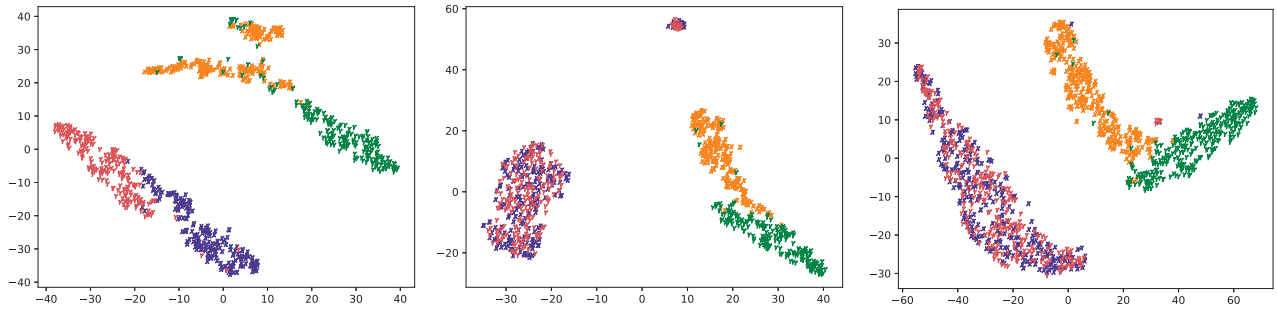
We further recorded the training time (s/batch) on SciTail on a single Nvidia Tesla V100 16 GB in Table III. The batch size of RE2-based, DecAtt-based, SBERT-based, and BERT-based models are set as 128, 128, 8, and 8 respectively. The results from Table III show that the computation time of G of JSD-based and MMD-based DDR-Match is far less than that of the underlying model. For WD-based G, it cost more time because of the alternative training, but the time cost of WD-based G is acceptable compared to the training time of the underlying model.

Summarizing the analysis above and the results reported in Table III, we can conclude that DDR-Match is an efficient framework that does not introduce many parameters compared to its underlying model.



(a) JSD-based regularizer (b) MMD-based regularizer (c) WD-based regularizer

Fig. 3. t-SNE visualization of the projected feature vectors, based on RE2 and DDR-Match (RE2) trained on SciTail. Subfigure (a), (b), and (c) respectively illustrate the vector distributions of DDR-Match(RE2) with JSD regularizer, MMD regularizer, and WD regularizer. The orange 'X' and green 'Y' correspond to $\mathbb{P}_F^X$ and $\mathbb{P}_F^Y$ of RE2, The dark blue 'X' and red 'Y' correspond to $\mathbb{P}_F^X$ and $\mathbb{P}_F^Y$ of DDR-Match (RE2), respectively.



(a) JSD-based regularizer (b) MMD-based regularizer (c) WD-based regularizer

Fig. 4. t-SNE visualization of the projected feature vectors, based on SBERT and DDR-Match (SBERT) trained on WikiQA. Subfigure (a), (b), and (c) respectively illustrate the vector distributions of DDR-Match(SBERT) with JSD regularizer, MMD regularizer, and WD regularizer. The orange 'X' and green 'Y' correspond to $\mathbb{P}_F^X$ and $\mathbb{P}_F^Y$ of SBERT, The dark blue 'X' and red 'Y' correspond to $\mathbb{P}_F^X$ and $\mathbb{P}_F^Y$ of DDR-Match (SBERT), respectively.

E. Visualization of the Distributions of Feature Vectors

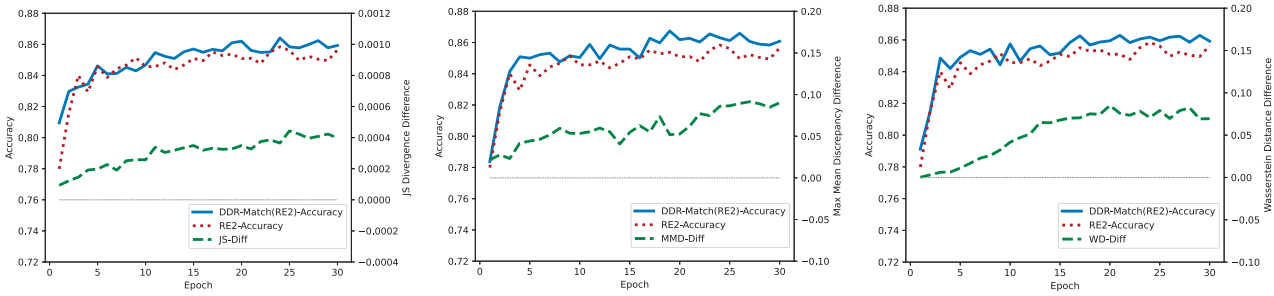
Fig. 1(a) shows that there exists a gap between two feature vectors, due to the heterogeneous nature of the texts from two asymmetrical domains. We conducted experiments to analyze how the feature vectors (i.e., $\mathbf{h}^X$ and $\mathbf{h}^Y$) generated by DDR-Match distributed in the common semantic space, taking DDR-Match (RE2) with and DDR-Match (SBERT) as examples. Specifically, we trained an RE2 model and a DDR-Match (RE2) model based on SciTail dataset. We recorded all of the training feature vectors and illustrated them in Fig. 3 by t-SNE. The orange 'X' and green 'Y' correspond to $\mathbb{P}_F^X$ and $\mathbb{P}_F^Y$ of RE2, The dark blue 'X' and red 'Y' correspond to $\mathbb{P}_F^X$ and $\mathbb{P}_F^Y$ of DDR-Match (RE2), respectively. As we can see from Fig. 3, the feature vectors from RE2 are separately distributed while the feature vectors from DDR-Match (RE2) are indistinguishable. We also trained an SBERT model and a DDR-Match (SBERT) model based on WikiQA dataset. Similar results can be found in Fig. 4. The feature vectors from SBERT are separately distributed while the feature vectors from DDR-Match (SBERT) are indistinguishable. These results demonstrate that compared to the underlying model, DDR-Match distributes the feature vectors in semantic space better and faster.

Please note that in Fig. 4, feature vectors of SBERT are separately distributed in the latent space (The orange 'X' and green 'Y'). It shows that there exists a larger gap on WikiQA dataset. Compared (a) to (b) and (c) of Fig. 4, we can see that the feature vectors are mixed more thoroughly with the guidance of MMD-based regularizer and WD-based regularizer than that of JSD-based regularizer. These results verified our assumption that when two distributions of sequence representations are disjoint, JSD-based regularizer failed to mix feature vectors effectively.

F. Convergence and Effects of WD-Based Regularizer

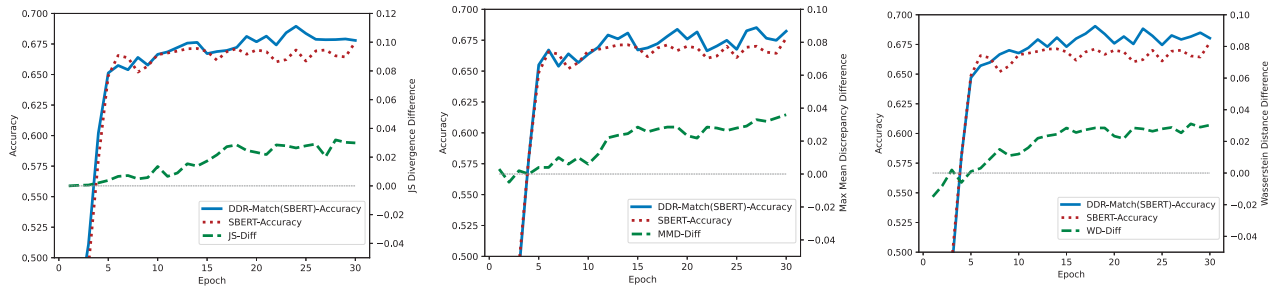
We conducted experiments to test how our proposed DDR-Match guided the training of matching models. Specifically, we compared DDR-Match and its underlying model under the guidance of the three types of regularizers on SciTail and WikiQA, respectively.

Taking the comparison between DDR-Match (RE2) and RE2 as an example, we tested the DDR-Match (RE2) and RE2 models generated at each training epochs. The accuracy curve on the basis of the development set of SciTail was illustrated in Fig. 5 (denoted as "DDR-Match (RE2)-Accuracy" and "RE2-Accuracy"). (a), (b), and (c) of Fig. 5 denotes DDR-Match based on JSD-based, MMD-based, and WD-based regularizer, respectively. Comparing these two training curves in (a), (b), and



(a) JSD-based regularizer (b) MMD-based regularizer (c) WD-based regularizer

Fig. 5. Subfigure (a), (b), and (c) are the accuracy curves and different distribution distance measurements curves w.r.t. training epochs for RE2 and DDR-Match (RE2) on Scitail.



(a) JSD-based regularizer (b) MMD-based regularizer (c) Wasserstein ditance-based regularizer

Fig. 6. Subfigure (a), (b), and (c) are the accuracy curves and different distribution distance measurements curve w.r.t. training epochs for SBERT and DDR-Match (SBERT) on WikiQA.

(c) of Fig. 5, we can see that DDR-Match (RE2) outperformed RE2 when the training closing to converge (after about 15 epochs). We can conclude that DDR-Match (RE2) obtained higher accuracy than RE2.

To investigate how the sequence representation distributions guide the training of matching models, we recorded the JSD, MMD, and the estimated WD at all of the training epochs of RE2 and DDR-Match (RE2). For example, in Fig. 5(c), the curve "WD-Diff" shows the differences between the WD by RE2 and that of by DDR-Match (RE2) at each of the training epoch (i.e., $\mathcal{L}_{wd}(\theta_F)$ of RE2 minus $\mathcal{L}_{wd}(\theta_F)$ of DDR-Match (RE2)). From the curve, we can see that at the beginning of the training (i.e., epoch 1 to 5), the "WD-Diff" was near to zero. With the training went on (i.e., epoch 5 to 30), the WD by DDR-Math(RE2) became smaller than that of by RE2 (the WD-Diff curve is above the zero line), which means that DDR-Match (RE2)'s feature projection module F was guided to move feature vectors together more thoroughly and faster, which are more suitable for matching. Similar phenomenon can be found in Fig. 5(a) and (c). These results indicate DDR-Match achieved its design goal of guiding the distributions of the projected feature vectors.

It is interesting to note that, comparing all of the three curves in Fig. 5(c), we found the WD-Diff curve is close to zero at the beginning of the training, and the accuracy curves of DDR-Match (RE2)-Accuracy and RE2-Accuracy are similar at the beginning. With the training went on (after epoch 10), the WD differences became larger. At the same time, the accuracy gaps (between DDR-Match (RE2)-Accuracy and RE2-Accuracy) also become larger. A similar phenomenon can be found in Fig. 5(a) and (c). The results clearly reflect the effects of sequence representation distributions distance-based regularizer: minimizing the regularizer leads to a better distribution of feature vectors in terms of matching. We also compared DDR-Match (SBERT) and SBERT on WikiQA as illustrated in Fig. 6. we can also obtain the same conclusion.

To further compare the effect of JSD-based, MMD-based, and WD-based regularizer. We illustrated the prediction accuracy and the estimated distribution distance at all of the training epochs of SBERT and DDR-Match (SBERT) in Fig. 6. Compared Fig. 6(a) to (b) and (c), DDR-Match (SBERT) with JSD-based regularizer converged after about 19 epochs of training, while DDR-Match (SBERT) with MMD-based or WD-based regularizer converged after about 15 epochs of training. Double-checking Fig. 4, the JSD regularizer failed to thoroughly mix the feature vectors of SBERT. These results verified our assumption that when two distributions of sequence representations are disjoint, JSD-based regularizer failed to mix feature vectors effectively.

G. *Effects of the Kernel Bandwidth in MMD-Based DDR-Match*

As the experimental results are shown in Table II DDR-Match outperformed its underlying models incorporated with the MMD-based regularizer in general. For the MMD-based regularizer, there is a key hyperparameter namely kernel bandwidth

TABLE IV
PERFORMANCE COMPARISON OF DIFFERENT SETTING ON KERNEL BANDWIDTH. DDR-MATCH(·) DENOTES DDR-MATCH WITH THE FIRST PARAMETER AS THE UNDERLYING MODEL, THE SECOND PARAMETER AS THE REGULARIZER

		SNLI	SICK	SciTail		WikiQA		TrecQA	
Models	**Bandwidth**	**Acc.**	**Acc.**	**Acc.**	**F1**	**MAP**	**MRR**	**MAP**	**MRR**
DDR-Match (RE2, MMD)	0.1	88.72	84.46	86.74	88.82	73.16	74.95	72.17	73.89
DDR-Match (RE2, MMD)	1.0	88.90	**85.39**	**87.39**	**89.58**	72.63	74.19	**73.76**	**75.30**
DDR-Match (RE2, MMD)	10.0	**89.02**	85.03	86.12	88.07	**74.84**	**76.89**	71.94	73.79
DDR-Match (DecAtt, MMD)	0.1	82.12	77.22	**83.49**	**86.23**	**65.04**	**66.87**	70.31	75.87
DDR-Match (DecAtt, MMD)	1.0	82.54	**78.11**	82.88	85.81	64.64	66.50	**70.88**	**76.36**
DDR-Match (DecAtt, MMD)	10.0	**82.78**	77.94	82.83	85.51	64.49	66.03	69.74	75.12
DDR-Match (SBERT, MMD)	0.1	83.20	**65.29**	80.36	83.35	**68.85**	**70.16**	**71.83**	**76.93**
DDR-Match (SBERT, MMD)	1.0	**83.91**	64.11	**81.67**	**84.49**	68.23	69.90	70.45	75.08
DDR-Match (SBERT, MMD)	10.0	83.67	63.64	80.88	83.78	68.47	70.12	71.36	76.10
DDR-Match (BERT, MMD)	0.1	**88.02**	86.04	89.60	91.60	78.84	80.95	78.84	80.95
DDR-Match (BERT, MMD)	1.0	87.85	**86.28**	**90.22**	**91.96**	79.11	81.06	78.11	80.06
DDR-Match (BERT, MMD)	10.0	87.95	86.12	89.89	91.77	**79.54**	**81.21**	**79.62**	**81.00**

TABLE V
PERFORMANCE COMPARISON OF REGULARIZATION ON ALL SAMPLES AND ON POSITIVE SAMPLES ON SNLI, SCITAIL, WIKIQA TEST SET. DDR-MATCH(·) DENOTES DDR-MATCH WITH THE FIRST PARAMETER AS THE UNDERLYING MODEL, THE SECOND PARAMETER AS THE REGULARIZER

		SNLI	SciTail		WikiQA	
Models	**Regularized samples**	**Acc.**	**Acc.**	**F1**	**MAP**	**MRR**
DDR-Match (RE2, JS)	positive	88.56	86.64	88.69	73.41	75.24
DDR-Match (RE2, JS)	all	**88.75**	**86.74**	**88.85**	**73.93**	**76.04**
DDR-Match (RE2, MMD)	positive	88.87	87.00	88.97	74.23	75.89
DDR-Match (RE2, MMD)	all	**89.02**	**87.39**	**89.58**	**74.84**	**76.24**
DDR-Match (RE2, WD)	positive	88.98	86.81	88.96	75.02	76.11
DDR-Match (RE2, WD)	all	**89.09**	**87.04**	**89.12**	**75.31**	**76.89**
DDR-Match (DecAtt, JS)	positive	82.45	81.57	84.43	64.16	66.00
DDR-Match (DecAtt, JS)	all	**82.56**	**82.07**	**84.86**	**64.67**	**66.77**
DDR-Match (DecAtt, MMD)	positive	82.52	82.23	85.45	64.20	66.08
DDR-Match (DecAtt, MMD)	all	**82.78**	**83.49**	**86.23**	**65.04**	**66.87**
DDR-Match (DecAtt, WD)	positive	82.49	82.14	85.17	64.33	66.25
DDR-Match (DecAtt, WD)	all	**82.62**	**82.94**	**85.88**	**65.16**	**67.24**

μ. In this subsection, we conducted experiments to investigate the effect of μ for MMD-based DDR-Match.

Different kernel bandwidth defines different reproducing kernel Hilbert spaces leading to different measurements to the distance of sequence representations. Thus, we performed an assessment for different values of the kernel bandwidth. Specifically, we chose $\mu \in \{0.1, 1, 10\}$ and conducted experiments to compare the performance of MMD-based DDR-Match on five datasets. Experimental results are listed in Table IV and all methods are trained five times and the average results are reported. Table IV clearly shows the effect of the kernel bandwidth. The performance of DDR-Match incorporated with MMD regularizer varied with different settings of μ. For example, DDR-Match (RE2) in the setting of $\mu = 10$ improved that of $\mu = 1$ about 3%. This result indicated that the kernel bandwidth has a major impact on the performance of the MMD-based regularizer. In practice, the kernel bandwidth of MMD-based DDR-Match needs to be carefully chosen.

H. Effects of Regularizing the Positive and All Samples

To further verify our idea, we conducted experiments to compare the performance of aligning the distribution of all sequence pairs and that of aligning the distribution of the positive sequence pairs. Specifically, on the task of NLI and QA, we adopted RE2 and DecAtt as the underlying matching methods of DDR-Match, used the JSD-based, MMD-based, and WD-based regularizer to respectively regularize only the positive sequence representations and all sequence representations. Experimental results are listed in Table V.

From the results, we found that on both NLI and QA tasks, DDR-Match that aligned the distribution of all sequence pairs performed better than aligned the distribution of the positive sequence pairs. We suppose the reason behind the results is as follows: 1) in asymmetrical text matching, the domain information contained in sequence representations is the noise for matching. In DDR-Match, aligning the distributions of the sequence pair can effectively denoise the information concerning domains while enforcing F to generate semantic-related representations that is helpful for the Matching Component M. 2) M of existing methods is capable of minimizing the distance between positive sequences representations and distinguishing the positive and the negative samples. When G is served as aligning the distribution of the positive sequence pairs, G plays a similar role to M. In this case, G may help M better distinguish between positive and negative samples, but G failed to remove the domain noise.

Summarizing the results and analysis above, we conclude that explicitly bridging the domain gap by aligning the distribution of sequence representations is beneficial for text matching in asymmetrical domains.

VI. Conclusion

In this paper, we observed that sequence representations tended to be indistinguishable in the latent space in the task of asymmetrical text matching. Inspired by this observation, we proposed a novel framework for asymmetrical text matching, namely DDR-Match which explicitly bridge the gap between sequence representations. To investigate the effectiveness of our proposed DDR-Match, we instantiated the JSD-based, MMD-based, and WD-based version of DDR-Match and incorporated four popular matching models into DDR-Match as its underlying models. We showed that the DDR-Match is capable of well distributing the generated feature vectors in the semantic space, and therefore more suitable for matching. Experimental results on five benchmarks showed that DDR-Match can outperform the baselines including its underlying models. Empirical analysis showed the effectiveness and the efficiency of DDR-Match and among the three instances of DDR-Match, the WD-based is more appealing for asymmetrical text matching.

References

[1] W. Yu *et al.*, "Wasserstein distance regularized sequence representation for text matching in asymmetrical domains," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2020, pp. 2985–2994.

[2] S. R. Bowman, G. Angeli, C. Potts, and C. D. Manning, "A large annotated corpus for learning natural language inference," in *Proc. Empirical Methods Natural Lang. Process.*, 2015, pp. 632–642.

[3] M. Wang, N. A. Smith, and T. Mitamura, "What is the jeopardy model? a quasi-synchronous grammar for QA," in *Proc. Joint Conf. Empirical Methods Natural Lang. Process. Computat. Natural Lang. Learn.*, 2007, pp. 22–32.

[4] Y. Yang, W.-t. Yih, and C. Meek, "WikiQA: A challenge dataset for open-domain question answering," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2015, pp. 2013–2018.

[5] P.-S. Huang, X. He, J. Gao, L. Deng, A. Acero, and L. Heck, "Learning deep structured semantic models for web search using clickthrough data," in *Proc. 22nd ACM Int. Conf. Inf. Knowl. Manage.*, 2013, pp. 2333–2338.

[6] A. Parikh, O. Täckström, D. Das, and J. Uszkoreit, "A decomposable attention model for natural language inference," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2016, pp. 2249–2255.

[7] R. Yang, J. Zhang, X. Gao, F. Ji, and H. Chen, "Simple and effective text matching with richer alignment features," in *Proc. 57th Annu. Meeting Assoc. Comput. Linguistics*, 2019, pp. 4699–4709, *arXiv:1908.00300*.

[8] N. Reimers and I. Gurevych, "Sentence-bert: Sentence embeddings using siamese bert-networks," in *Proc. Conf. Empirical Methods Natural Lang. Process. 9th Int. Joint Conf. Natural Lang. Process.*, 2019, pp. 3973–3983.

[9] T. Khot, A. Sabharwal, and P. Clark, "Scitail: A textual entailment dataset from science question answering," in *Proc. AAAI Conf. Artif. Intell.*, 2018, vol. 17, pp. 41–42.

[10] L. v. d. Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, no. 11, pp. 2579–2605, 2008.

[11] Y. Tay, A. T. Luu, and S. C. Hui, "Hermitian co-attention networks for text matching in asymmetrical domains," in *Proc. Int. Joint Conf. Artif. Intell.*, 2018, pp. 4425–4431.

[12] B. Wang, Y. Yang, X. Xu, A. Hanjalic, and H. T. Shen, "Adversarial cross-modal retrieval," in *Proc. 25th ACM Int. Conf. Multimedia*, 2017, pp. 154–162.

[13] K. Leidal, D. Harwath, and J. Glass, "Learning modality-invariant representations for speech and images," in *Proc. IEEE Autom. Speech Recognit. Understanding Workshop*, 2017, pp. 424–429.

[14] J. Li, M. Jing, L. Zhu, Z. Ding, K. Lu, and Y. Yang, "Learning modality-invariant latent representations for generalized zero-shot learning," in *Proc. 28th ACM Int. Conf. Multimedia*, 2020, pp. 1348–1356.

[15] D. Hazarika, R. Zimmermann, and S. Poria, "Misa: Modality-invariant and-specific representations for multimodal sentiment analysis," in *Proc. 28th ACM Int. Conf. Multimedia*, 2020, pp. 1122–1131.

[16] Y. Shen, X. He, J. Gao, L. Deng, and G. Mesnil, "A latent semantic model with convolutional-pooling structure for information retrieval," in *Proc. 23rd ACM Int. Conf. Conf. Inf. Knowl. Manage.*, 2014, pp. 101–110.

[17] B. Hu, Z. Lu, H. Li, and Q. Chen, "Convolutional neural network architectures for matching natural language sentences," in *Adv. Neural Inf. Process. Syst.*, 2014, pp. 2042–2050.

[18] X. Qiu and X. Huang, "Convolutional neural tensor network architecture for community-based question answering," in *Proc. 24th Int. Conf. Artif. Intell.*, 2015, pp. 1305–1311.

[19] H. Palangi *et al.*, "Deep sentence embedding using long short-term memory networks: Analysis and application to information retrieval," *IEEE/ACM Trans. Audio, Speech Lang. Process.*, vol. 24, no. 4, pp. 694–707, Apr. 2016.

[20] S. Wan, Y. Lan, J. Guo, J. Xu, L. Pang, and X. Cheng, "A deep architecture for semantic matching with multiple positional sentence representations," *Proc. AAAI Conf. Artif. Intell.*, 2016, vol. 30, no. 1, *arXiv:1511.08277*.

[21] Y. Tay, A. T. Luu, and S. C. Hui, "Co-stack residual affinity networks with multi-level attention refinement for matching text sequences," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, Brussels, Belgium, Association for Computational Linguistics, 2018, pp. 4492–4502.

[22] S. Kim, I. Kang, and N. Kwak, "Semantic sentence matching with densely-connected recurrent and co-attentive information," *Proc. AAAI Conf. Artif. Intell.*, 2019, vol. 33, no. 1, pp. 6586–6593.

[23] Y. Zhang, J. Zhang, Z. Cui, S. Wu, and L. Wang, "A graph-based relevance matching model for ad-hoc retrieval," in *Proc. AAAI Conf. Artif. Intell.*, 2021, vol. 35, no. 5, pp. 4688–4696.

[24] S. Peng, H. Cui, N. Xie, S. Li, J. Zhang, and X. Li, "Enhanced-RCNN: An efficient method for learning sentence similarity," in *Proc. The Web Conf.*, 2020, pp. 2500–2506.

[25] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *Proc. 2019 Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, 2019, vol. 1, pp. 4171–4186, *arXiv:1810.04805*.

[26] B. Fuglede and F. Topsoe, "Jensen-Shannon divergence and Hilbert space embedding," in *Proc. IEEE Int. Symp. Inf. Theory*, Chicago, IL, USA: 2004, p. 31.

[27] R. J. Gallagher, A. J. Reagan, C. M. Danforth, and P. S. Dodds, "Divergent discourse between protests and counter-protests:# blacklivesmatter and# alllivesmatter," *PLoS one*, vol. 13, no. 4, 2018, Art. no. e0 195644.

[28] J. Lu, M. Henchion, and B. MacNamee, "Extending Jensen Shannon divergence to compare multiple corpora," in *Proc. 25th Irish Conf. Artif. Intell. Cogn. Sci.*, Dublin, Ireland, Dec. 2017. [Online]. Available: http://Ceur-ws.org

[29] E. A. Pechenick, C. M. Danforth, and P. S. Dodds, "Characterizing the google books corpus: Strong limits to inferences of socio-cultural and linguistic evolution," *PLoS One*, vol. 10, no. 10, 2015, Art. no. e0137041.

[30] A. Koplenig, "A fully data-driven method to identify (correlated) changes in diachronic corpora," *CoRR*, 2015, *arXiv:1508.06374*.

[31] R. Zafarani and H. Liu, "Connecting users across social media sites: A behavioral-modeling approach," in *Proc. 19th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2013, Art. no. 4149. [Online]. Available: https://doi.org/10.1145/2487575.2487648

[32] Y. Li, Y. Peng, Z. Zhang, M. Wu, Q. Xu, and H. Yin, "A deep dive into user display names across social networks," *Inf. Sci.*, vol. 447, pp. 186–204, 2018.

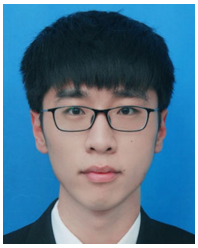
[33] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola, "A kernel two-sample test," *J. Mach. Learn. Res.*, vol. 13, no. 1, pp. 723–773, 2012.

[34] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, and T. Darrell, "Deep domain confusion: Maximizing for domain invariance," *CoRR*, 2014, *arXiv:1412.3474*.

[35] M. Long, Y. Cao, J. Wang, and M. Jordan, "Learning transferable features with deep adaptation networks," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 97–105.

[36] M. Long, H. Zhu, J. Wang, and M. I. Jordan, "Deep transfer learning with joint adaptation networks," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 2208–2217.

[37] J. You, R. Ying, X. Ren, W. Hamilton, and J. Leskovec, "Graphrnn: Generating realistic graphs with deep auto-regressive models," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 5708–5717.
[38] B. Tran, M. Karimzadehgan, R. K. Pasumarthi, M. Bendersky, and D. Metzler, "Domain adaptation for enterprise email search," in *Proc. 42nd Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2019, pp. 25–34.
[39] X. Chen, Y. Sun, B. Athiwaratkun, C. Cardie, and K. Weinberger, "Adversarial deep averaging networks for cross-lingual sentiment classification," *Trans. Assoc. Comput. Linguist.*, vol. 6, pp. 557–570, 2018.
[40] I. Goodfellow *et al.*, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.
[41] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 214–223.
[42] Y. Ganin *et al.*, "Domain-adversarial training of neural networks," *J. Mach. Learn. Res.*, vol. 17, no. 1, pp. 2096–2030, 2016.
[43] I. O. Tolstikhin, O. Bousquet, S. Gelly, and B. Schölkopf, "Wasserstein auto-encoders," in *Proc. 6th Int. Conf. Learn. Representations*, Vancouver, BC, Canada, 2018, *arXiv:1711.01558*.
[44] D. Zhu, P. Cui, D. Wang, and W. Zhu, "Deep variational network embedding in wasserstein space," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Data Mining*, 2018, pp. 2827–2836.
[45] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola, "A kernel method for the two-sample problem," *J. Mach. Learn. Res.*, vol. 1, pp. 1–10, 2008.
[46] C. Villani, "*Topics in Optimal Transportation*," *Amer. Math. Soc.*, 2003.
[47] L. Zhang, S. R. Wilson, and R. Mihalcea, "Multi-label transfer learning for multi-relational semantic similarity," in *Proc. 8th Joint Conf. Lexical Comput. Semantics*, Minneapolis, MN, USA, 2019, pp. 44–50.
[48] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2014, pp. 1532–1543.
[49] M. Arjovsky and L. Bottou, "Towards principled methods for training generative adversarial networks," in *5th Int. Conf. Learn. Representations*, Toulon, France, Apr. 2017, *arXiv:1701.04862*.



Weijie Yu received the M.S. degree from the University of Science and Technology of China, Hefei, China, in 2018. He is currently working toward the Ph.D. degree with School of Information, Renmin University of China, Beijing, China. His research interests include text matching and legal retrieval.

Chen Xu is currently working toward the Ph.D. degrees with Gaoling Artificial Intelligence School, Renmin University of China, Beijing, China, and the University of Montreal, Montreal, QC, Canada.

His current research interests mainly include the causal inference for natural language processing and recommendation tasks.

Jun Xu is currently a Professor with the Gaoling School of Artificial Intelligence, Renmin University of China, Beijing, China.

He has authored or coauthored more than 70 papers in international conferences, such as SIGIR and WWW and journals, including *ACM Transactions on Information Systems*, IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING. His research interests include learning to rank and semantic matching in web search. He was/is a SPC of the SIGIR, WWW, and AAAI, an Editorial Board Member of the JASIST, and an Associate Editor for the TIST.

He was the recipient of the Test of Time Award Honorable Mention in SIGIR (2019), Best Paper Award in AIRS (2010) and Best Paper Runner-up in CIKM (2017).

Liang Pang is currently an Associate Professor with the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China. He has authored or coauthored about 30 papers in top international journals and conferences, including SIGIR, CIKM, ACL, AAAI, and IJCAI. His research interests include designing deep models for text matching and learning-to-rank in information retrieval.

His work on information retrieval was the recipient of the Best Paper Runner-up of ACM CIKM 2017. He is very active in the research communities. He was/is a PC Member of top international conferences, including SIGIR, WWW, NeurIPS, AAAI, IJCAI, and CIKM.

Ji-Rong Wen is currently a Professor with the Renmin University of China (RUC), Beijing, China. He is also the Dean of School of Information and the Executive Dean of Gaoling School of Artificial Intelligence, RUC. His main research interests include information retrieval, data mining, and machine learning.

He was a Senior Researcher and Group Manager of the Web Search and Mining Group, Microsoft Research Asia.