Dual-View Variational Autoencoders for Semi-Supervised Text Matching

Zhongbin Xie and **Shuai Ma**
SKLSDE Lab, Beihang University, Beijing, China
Beijing Advanced Innovation Center for Big Data and Brain Computing, Beijing, China
{xiezb, mashuai}@buaa.edu.cn

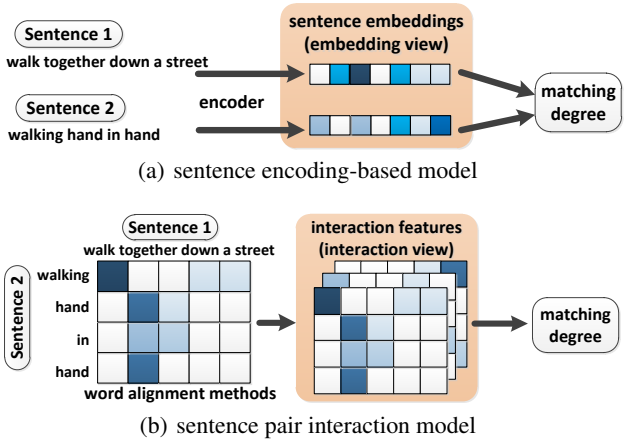
Abstract

Semantically matching two text sequences (usually two sentences) is a fundamental problem in NLP. Most previous methods either encode each of the two sentences into a vector representation (sentence-level embedding) or leverage word-level interaction features between the two sentences. In this study, we propose to take the sentence-level embedding features and the word-level interaction features as two distinct views of a sentence pair, and unify them with a framework of Variational Autoencoders such that the sentence pair is matched in a semi-supervised manner. The proposed model is referred to as Dual-View Variational AutoEncoder (DV-VAE), where the optimization of the variational lower bound can be interpreted as an implicit Co-Training mechanism for two matching models over distinct views. Experiments on SNLI, Quora and a Community Question Answering dataset demonstrate the superiority of our DV-VAE over several strong semi-supervised and supervised text matching models.

1 Introduction

The need to semantically match two text sequences arises in many Natural Language Processing problems, where a central task is to compute the matching degree between two sentences and determine their semantic relationship. For instance, in Paraphrase Identification [Dolan and Brockett, 2005], whether one sentence is a paraphrase of another has to be determined; In Question Answering [Yang *et al.*, 2015], a matching score is calculated for a question and each of its candidate answers for making decisions; And in Natural Language Inference [Bowman *et al.*, 2015], the relationship between a premise and a hypothesis is classified as entailment, neutral or contradiction.

Most previous studies on text matching focus on developing supervised models with deep neural networks. These models can be essentially divided into two categories: (i) *sentence encoding-based models*, which separately encode each of the two sentences into a vector representation (sentence embedding) and then match between the two vectors [Bowman *et al.*, 2016a; Mueller and Thyagarajan, 2016], and (ii)



(a) sentence encoding-based model

(b) sentence pair interaction model

Figure 1: Two categories of text matching models: They leverage sentence embeddings and word-level interaction features, respectively, which can be seen as two distinct views of the sentence pair.

sentence pair interaction models, which use some sorts of word alignment methods, such as interaction matrices [Gong *et al.*, 2018; Wu *et al.*, 2018] or attention mechanisms [Rocktäschel *et al.*, 2016; Wang and Jiang, 2017], to obtain fine-grained interaction features for predicting the matching degree. Sentence encoding-based models leverage global sentence representations with high-level semantic features, while sentence pair interaction models leverage word-by-word interaction features containing local matching patterns, as illustrated in Figure 1.

With the recent advances in deep generative models, some studies begin to employ variational autoencoders (VAEs) [Kingma and Welling, 2014] to learn informative sentence embeddings for various downstream NLP problems, including text matching [Bowman *et al.*, 2016b; Zhao *et al.*, 2018; Shen *et al.*, 2018]. They leverage a VAE to encode sentences into latent codes, which are used as sentence embeddings for a sentence encoding-based matching model. The VAE and the matching model can be jointly trained in a semi-supervised manner, leveraging large amounts of unlabeled data to improve matching performance. *However, these models are limited to global semantic features in the sentence embeddings, leaving out the word-level interaction features that*

have been proved important for predicting matching degrees in the supervised case [Lan and Xu, 2018].

Motivated by these observations, we propose to unify the sentence-level embedding features and the word-level interaction features within a variational autoencoder, leveraging both labeled and unlabeled sentence pairs in a semi-supervised manner for text matching. We take inspiration from Co-Training [Blum and Mitchell, 1998], where two classifiers over two distinct views of the data examples are trained to produce consistent predictions on the unlabeled data. For a sentence pair, the aforementioned two levels of features are taken as two distinct views, namely the *embedding view* and the *interaction view*. The proposed model is denoted Dual-View Variational AutoEncoder (DV-VAE) (Figure 2). In the generative process, two sentences are generated from two latent variables, respectively. The matching degree is also generated from these two latent variables, treated as the embedding view, through a *sentence encoding-based model*. In the inference process, the matching degree is inferred from the interaction view through a *sentence pair interaction model*. During the optimization of the variational lower bound, the two matching models implicitly provide pseudo labels on unlabeled data for each other, which can be interpreted as an implicit Co-Training mechanism.

Our contributions are as follows: (i) We propose Dual-View Variational AutoEncoder (DV-VAE) to unify the embedding view and the interaction view of a sentence pair for semi-supervised text matching. An implicit Co-Training mechanism is also formulated to interpret the training process. (ii) We instantiate an implementation of DV-VAE and adopt a novel sentence pair interaction matching model, where interaction matrices across words and contexts are introduced to enrich the interaction features. (iii) Using three datasets: SNLI, Quora and a Community QA dataset, we empirically demonstrate the superiority of DV-VAE over several strong semi-supervised and supervised baselines.

2 Dual-View Variational Antoencoder

Suppose that we have a labeled sentence pair set $\mathbb{D}_l$ and an unlabeled sentence pair set $\mathbb{D}_u$. $(x_1, x_2, y) \in \mathbb{D}_l$ denotes a labeled sentence pair, where x_1, x_2 are two sentences and $y \in \{1, 2, \ldots, C\}$ is the matching degree of x_1 and x_2. Here y is discretized and text matching is treated as a classification problem. Similarly, $(x_1, x_2) \in \mathbb{D}_u$ denotes an unlabeled pair. Our goal is to develop a semi-supervised text matching model using both the labeled and unlabeled data $\mathbb{D}_l$ and $\mathbb{D}_u$, which can improve upon the performance of supervised text matching models using the labeled data $\mathbb{D}_l$ only.

2.1 Model Architecture

The probabilistic graphical model of DV-VAE is shown in Figure 2. It consists of a generative model matching from the embedding view and an inference model matching from the interaction view.

Generative Model. The generative process of a sentence pair and their matching degree (x_1, x_2, y) is defined as follows: two continuous latent codes $\mathbf{z}_1, \mathbf{z}_2 \in \mathbb{R}^{d_z}$ are independently sampled from a prior $p(\mathbf{z})$, and are used to generate

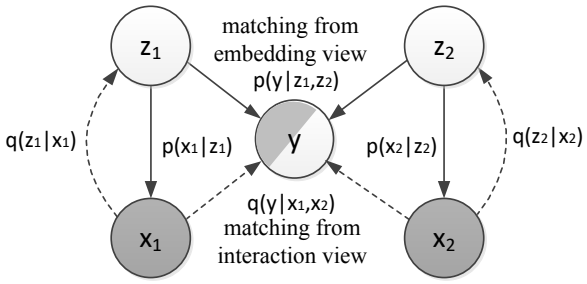


Figure 2: Probabilistic graphical model of DV-VAE. Solid lines denote the generative model, and dashed lines denote the inference model. Shaded x_1, x_2 are observed variables; $\mathbf{z}_1, \mathbf{z}_2$ are latent variables; and y is a semi-observed variable.

x_1 and x_2 through a decoder $p_\theta(x|\mathbf{z})$. Latent codes $\mathbf{z}_1, \mathbf{z}_2$ are also fed into a sentence encoding-based matching model $p_\theta(y|\mathbf{z}_1, \mathbf{z}_2)$ to generate the matching degree y. The joint distribution can be explained by the following factorization:

$$p_\theta(x_1, x_2, y, \mathbf{z}_1, \mathbf{z}_2) = p(\mathbf{z}_1)p_\theta(x_1|\mathbf{z}_1)p(\mathbf{z}_2)p_\theta(x_2|\mathbf{z}_2)p_\theta(y|\mathbf{z}_1, \mathbf{z}_2),$$

where $p(\mathbf{z}_1) = p(\mathbf{z}_2) = p(\mathbf{z}) = \mathcal{N}(\mathbf{z}; \mathbf{0}, \mathbf{I})$ is a Gaussian prior. And $p_\theta(y|\mathbf{z}_1, \mathbf{z}_2)$ is referred to as the embedding matcher as it matches from the embedding view (latent space).

Inference Model. According to the conditional independence properties in the generative model, the variational posterior $q_\phi(\mathbf{z}_1, \mathbf{z}_2, y|x_1, x_2)$ can be factorized as:

$$\begin{aligned} q_\phi(\mathbf{z}_1, \mathbf{z}_2, y|x_1, x_2) &= q_\phi(\mathbf{z_1}|x_1)q_\phi(\mathbf{z_2}|x_2)q_\phi(y|\mathbf{z}_1, \mathbf{z}_2) \\ &= q_\phi(\mathbf{z_1}|x_1)q_\phi(\mathbf{z_2}|x_2)p_\theta(y|\mathbf{z}_1, \mathbf{z}_2), \quad (1) \end{aligned}$$

where we model $q_\phi(y|\mathbf{z}_1, \mathbf{z}_2)$ by the embedding matcher $p_\theta(y|\mathbf{z}_1, \mathbf{z}_2)$. $q_\phi(\mathbf{z}_1, \mathbf{z}_2, y|x_1, x_2)$ can also be factorized as:

$$q_\phi(\mathbf{z}_1, \mathbf{z}_2, y|x_1, x_2) = q_\phi(y|x_1, x_2)q_\phi(\mathbf{z_1}, \mathbf{z_2}|x_1, x_2, y), \quad (2)$$

where we model $q_\phi(y|x_1, x_2)$ by a sentence pair interaction matching model to match from the interaction view. Thus $q_\phi(y|x_1, x_2)$ is referred to as the interaction matcher and is adopted to make predictions at test time. In analogy to Co-Training [Blum and Mitchell, 1998], we assume that each of the embedding view and the interaction view is sufficient to train the corresponding matcher, and the predictions from the two matchers are consistent in the inference process: $q_\phi(y|x_1, x_2) = p_\theta(y|\mathbf{z}_1, \mathbf{z}_2)$. With this consistency assumption, we obtain the following from Equ (1) and Equ (2):

$$\begin{aligned} q_\phi(\mathbf{z_1}, \mathbf{z_2}|x_1, x_2, y) &= q_\phi(\mathbf{z_1}|x_1)q_\phi(\mathbf{z_2}|x_2), \\ q_\phi(\mathbf{z}_1, \mathbf{z}_2, y|x_1, x_2) &= q_\phi(y|x_1, x_2)q_\phi(\mathbf{z_1}|x_1)q_\phi(\mathbf{z_2}|x_2), \end{aligned}$$

which are taken as the inference model in labeled and unlabeled cases, respectively. Here encoders $q_\phi(\mathbf{z_1}|x_1) = \mathcal{N}(\mathbf{z}_1; \mu_\phi(x_1), \text{diag}(\sigma^2_\phi(x_1)))$ and $q_\phi(\mathbf{z_2}|x_2) = \mathcal{N}(\mathbf{z}_2; \mu_\phi(x_2), \text{diag}(\sigma^2_\phi(x_2)))$ are diagonal Gaussians.

Objective

The variational lower bound of the data likelihood is used as the objective, for both the labeled and unlabeled data.

For a labeled sentence pair (x_1, x_2, y),

$$\begin{aligned}
&\log p_\theta(x_1, x_2, y) \\
&\geq \mathbb{E}_{q_\phi(\mathbf{z}_1,\mathbf{z}_2|x_1,x_2,y)}\left[\log \frac{p_\theta(x_1, x_2, y, \mathbf{z}_1, \mathbf{z}_2)}{q_\phi(\mathbf{z}_1, \mathbf{z}_2|x_1, x_2, y)}\right] \\
&= \mathbb{E}_{q_\phi(\mathbf{z}_1|x_1)q_\phi(\mathbf{z}_2|x_2)}[\log p_\theta(x_1|\mathbf{z}_1) + \\
&\quad \log p_\theta(x_2|\mathbf{z}_2) + \log p_\theta(y|\mathbf{z}_1, \mathbf{z}_2)] \\
&\quad - \mathrm{KL}(q_\phi(\mathbf{z}_1|x_1)\|p(\mathbf{z}_1)) - \mathrm{KL}(q_\phi(\mathbf{z}_2|x_2)\|p(\mathbf{z}_2)) \\
&\equiv -\mathcal{L}(x_1, x_2, y).
\end{aligned}$$

We rewrite $\mathcal{L}(x_1, x_2, y)$ as

$$\begin{aligned}
\mathcal{L}(x_1, x_2, y) &= -\mathcal{R}(x_1, x_2) - \mathcal{D}(x_1, x_2, y) \\
&\quad + \mathrm{KL}(q_\phi(\mathbf{z}_1|x_1)\|p(\mathbf{z}_1)) + \mathrm{KL}(q_\phi(\mathbf{z}_2|x_2)\|p(\mathbf{z}_2)),
\end{aligned} \tag{3}$$

where $-\mathcal{R}(x_1, x_2) = \mathbb{E}_{q_\phi(\mathbf{z}_1|x_1)q_\phi(\mathbf{z}_2|x_2)}[-\log p_\theta(x_1|\mathbf{z}_1) - \log p_\theta(x_2|\mathbf{z}_2)]$ is the reconstruction loss of x_1 and x_2; $-\mathcal{D}(x_1, x_2, y) = \mathbb{E}_{q_\phi(\mathbf{z}_1|x_1)q_\phi(\mathbf{z}_2|x_2)}[-\log p_\theta(y|\mathbf{z}_1, \mathbf{z}_2)]$ can be seen as an expected discriminative loss for the embedding matcher $p_\theta(y|\mathbf{z}_1, \mathbf{z}_2)$; and the last two KL-divergence terms regularize the posteriors to be close to the priors.

For an unlabeled sentence pair (x_1, x_2),

$$\begin{aligned}
&\log p_\theta(x_1, x_2) \\
&\geq \mathbb{E}_{q_\phi(\mathbf{z}_1,\mathbf{z}_2,y|x_1,x_2)}\left[\log \frac{p_\theta(x_1, x_2, y, \mathbf{z}_1, \mathbf{z}_2)}{q_\phi(\mathbf{z}_1, \mathbf{z}_2, y|x_1, x_2)}\right] \\
&= \mathbb{E}_{q_\phi(y|x_1,x_2)}\left[\mathbb{E}_{q_\phi(\mathbf{z}_1,\mathbf{z}_2|x_1,x_2,y)}\left[\log \frac{p_\theta(x_1, x_2, y, \mathbf{z}_1, \mathbf{z}_2)}{q_\phi(\mathbf{z}_1, \mathbf{z}_2|x_1, x_2, y)}\right]\right. \\
&\quad \left. - \log q_\phi(y|x_1, x_2)\right] \\
&= \sum_y q_\phi(y|x_1, x_2)(-\mathcal{L}(x_1, x_2, y)) + \mathcal{H}[q_\phi(y|x_1, x_2)] \\
&\equiv -\mathcal{U}(x_1, x_2).
\end{aligned} \tag{4}$$

Since $q_\phi(y|x_1, x_2)$ is not included in the expression of $\mathcal{L}(x_1, x_2, y)$, we explicitly add a discriminative loss for $q_\phi(y|x_1, x_2)$, weighted by α:

$$\mathcal{L}^\alpha(x_1, x_2, y) = \mathcal{L}(x_1, x_2, y) + \alpha[-\log q_\phi(y|x_1, x_2)]. \tag{5}$$

Finally, we obtain the objective function to be minimized on the entire dataset $\mathbb{D}_l \cup \mathbb{D}_u$:

$$\mathcal{J} = \sum_{(x_1,x_2,y)\in\mathbb{D}_l} \mathcal{L}^\alpha(x_1, x_2, y) + \sum_{(x_1,x_2)\in\mathbb{D}_u} \mathcal{U}(x_1, x_2). \tag{6}$$

2.2 Implicit Co-Training

In this section, we show that the training process of DV-VAE is implicitly related to Co-Training [Blum and Mitchell, 1998], where two classifiers are iteratively trained to explicitly provide pseudo labels on unlabeled data for each other. Since in DV-VAE the embedding matcher $p_\theta(y|\mathbf{z}_1, \mathbf{z}_2)$ and the interaction matcher $q_\phi(y|x_1, x_2)$ are simultaneously trained through the optimization of $\mathcal{J}$ in Equ (6), we analyze their gradients to study the training process. For clarity, we specify the parameters in $q_\phi(y|x_1, x_2)$ as ϕ_m and the parameters in $p_\theta(y|\mathbf{z}_1, \mathbf{z}_2)$ as θ_m, respectively.

(i) For a labeled sentence pair, minimizing $\mathcal{L}^\alpha(x_1, x_2, y)$ also minimizes the discriminative losses ($-\mathcal{D}(x_1, x_2, y)$ and $-\log q_\phi(y|x_1, x_2)$) for the two matchers, which are independently trained just as in supervised learning.

(ii) For an unlabeled sentence pair, we analyze the gradients of $\mathcal{U}(x_1, x_2)$ in Equ (4) w.r.t. θ_m and ϕ_m, respectively.

For the embedding matcher $p_{\theta_m}(y|\mathbf{z}_1, \mathbf{z}_2)$,

$$\nabla_{\theta_m}\mathcal{U}(x_1, x_2) = \sum_y q_{\phi_m}(y|x_1, x_2)\nabla_{\theta_m}[-\mathcal{D}(x_1, x_2, y)], \tag{7}$$

where the discriminative gradient $\nabla_{\theta_m}[-\mathcal{D}(x_1, x_2, y)]$ is reweighted by the predicted distribution $q_{\phi_m}(y|x_1, x_2)$ from the interaction matcher.

For the interaction matcher $q_{\phi_m}(y|x_1, x_2)$,[1]

$$\begin{aligned}
&\nabla_{\phi_m}\mathcal{U}(x_1, x_2) \\
&= \sum_y \mathcal{L}(x_1, x_2, y)\nabla_{\phi_m}[q_{\phi_m}(y|x_1, x_2)] \\
&\quad - \nabla_{\phi_m}\mathcal{H}[q_{\phi_m}(y|x_1, x_2)] \qquad (8) \\
&= \sum_y \big(-\mathcal{D}(x_1, x_2, y)\big)\nabla_{\phi_m}[q_{\phi_m}(y|x_1, x_2)] \\
&\quad - \nabla_{\phi_m}\mathcal{H}[q_{\phi_m}(y|x_1, x_2)] \qquad (9) \\
&= \sum_y q_{\phi_m}(y|x_1, x_2)\Big\{\mathcal{D}(x_1, x_2, y)\nabla_{\phi_m}[\\
&\quad - \log q_{\phi_m}(y|x_1, x_2)]\Big\} - \nabla_{\phi_m}\mathcal{H}[q_{\phi_m}(y|x_1, x_2)]. \qquad (10)
\end{aligned}$$

The first term can be seen as an application of the REINFORCE algorithm [Williams, 1992] from Reinforcement Learning, such that $\mathcal{D}(x_1, x_2, y)$, matching degree y, sentence pair (x_1, x_2) and $q_{\phi_m}(y|x_1, x_2)$ correspond to the reward signal, action, state and decision model, respectively [Mnih and Gregor, 2014; Xu *et al.*, 2017]. The second term maximizes the entropy of $q_{\phi_m}(y|x_1, x_2)$, and is treated as a regularizer.

With the training process going on, the two matchers distinguish correct and incorrect labels better and better through the supervised loss $\mathcal{L}^\alpha(x_1, x_2, y)$. Therefore, for unlabeled sentence pairs, the weight for the embedding matcher's discriminative gradient in Equ (7) becomes larger on correct ys, and the interaction matcher receives larger reward signals when it gives correct predictions in Equ (10). This is an alternative way of providing pseudo labels for unlabeled data, and can be treated as an implicit Co-Training mechanism.

2.3 Model Implementation

We present an implementation of DV-VAE (shown in Figure 3) in detail, which consists of an encoder $q_\phi(\mathbf{z}|x)$, a decoder $p_\theta(x|\mathbf{z})$, an embedding matcher $p_\theta(y|\mathbf{z}_1, \mathbf{z}_2)$ and an interaction matcher $q_\phi(y|x_1, x_2)$.

[1] To derive Equ (8), note that ϕ_m does not exist in $\mathcal{L}(x_1, x_2, y)$; to derive Equ (9), note that $\sum_y K\nabla_\omega q(y;\omega) = K\nabla_\omega \sum_y q(y;\omega) = K\nabla_\omega 1 = \mathbf{0}$ when K is irrelevant with y, which is the case for $\mathcal{R}(x_1, x_2)$ and the KL terms in $\mathcal{L}(x_1, x_2, y)$; to derive Equ (10), use $\nabla_\omega q(y;\omega) = q(y;\omega)\nabla_\omega \log q(y;\omega)$.

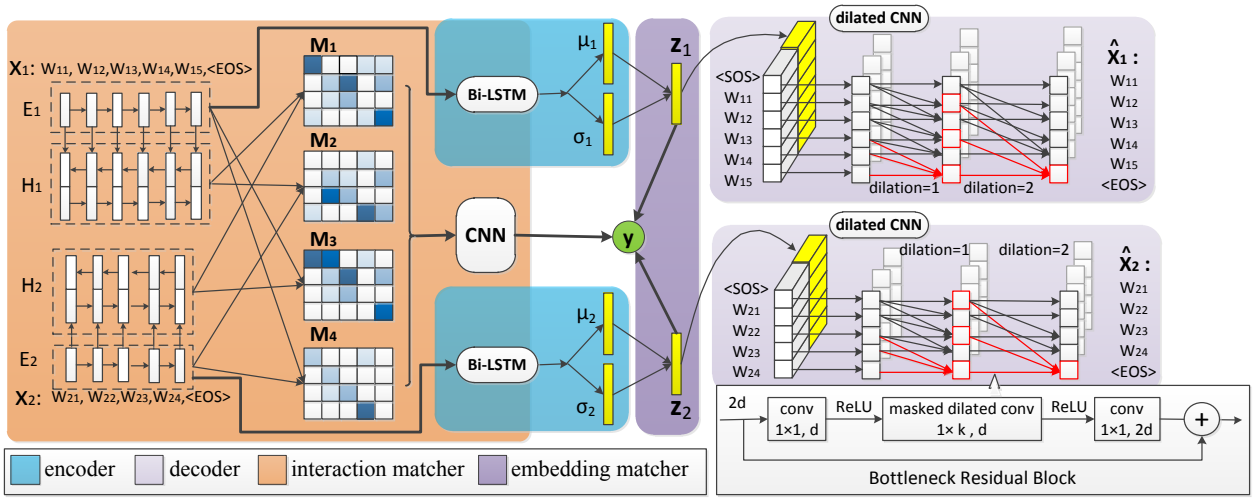


Figure 3: An implementation of DV-VAE.

Encoder $q_\phi(\mathbf{z}|x)$**.** We adopt a bidirectional LSTM (Bi-LSTM) as the encoder. The last hidden states of the forward and backward directions are concatenated and fed into two Multi-Layer Perceptrons (MLPs) to compute the mean μ and the standard deviation σ for $q_\phi(\mathbf{z}|x) = \mathcal{N}(\mathbf{z}; \mu, \text{diag}(\sigma^2))$.

Decoder $p_\theta(x|\mathbf{z})$**.** To avoid the training collapse of VAE-based text generation models, we adopt a dilated CNN sequence decoder that is most similar to the one in [Yang *et al.*, 2017]: Latent code $\mathbf{z}$ is concatenated to every word embedding of x to serve as the decoder input. Every feature dimension of the decoder input is treated as a channel and *masked one-dimensional convolution* proceeds in the time dimension. *Dilated convolution* with rate r is applied so that one in every r consecutive inputs is picked to convolve with the filter. Multiple dilated convolution layers are stacked with exponentially increasing dilation rates $[2^0, 2^1, 2^2, \dots]$, and every layer is wrapped in a *bottleneck residual block* [He *et al.*, 2016] to ease optimization. The outputs of the last layer are fed into a fully connected layer followed by a softmax nonlinearity to produce the probability $p_\theta(w_j|w_{<j}, \mathbf{z})$ for all time steps $j \in [1, \dots, T]$. Then the reconstruction probability of text sequence x is computed as $p_\theta(x|\mathbf{z}) = \prod_{j=1}^{T} p_\theta(w_j|w_{<j}, \mathbf{z})$.

Embedding matcher $p_\theta(y|\mathbf{z}_1, \mathbf{z}_2)$**.** We adopt an MLP taking as input the concatenation of $\mathbf{z}_1, \mathbf{z}_2$, the element-wise difference $\mathbf{z}_1 - \mathbf{z}_2$ and the element-wise product $\mathbf{z}_1 \odot \mathbf{z}_2$:

$$p_\theta(y|\mathbf{z}_2, \mathbf{z}_2) = \text{softmax}(\text{MLP}([\mathbf{z}_1; \mathbf{z}_2; \mathbf{z}_1 - \mathbf{z}_2; \mathbf{z}_1 \odot \mathbf{z}_2]))$$

Interaction matcher $q_\phi(y|x_1, x_2)$**.** For $x_1 = \{w_{11}, w_{12}, \dots, w_{1T_1}\}$, we denote the word embedding sequence as $E_1 = \{e_{11}, e_{12}, \dots, e_{1T_1}\}$,where $e_{1j} \in \mathbb{R}^{d_E}$ is the word embedding for token w_{1j}. Then a bidirectional LSTM (Bi-LSTM) is adopted to get a context sequence $H_1 = \{h_{11}, h_{12}, \dots, h_{1T_1}\}$, where $h_{1j} = [\overrightarrow{h_{1j}}; \overleftarrow{h_{1j}}]$ is the concatenation of the corresponding forward and backward hidden states of the Bi-LSTM, and $\overrightarrow{h_{1j}}, \overleftarrow{h_{1j}} \in \mathbb{R}^{d_H}$. Similarly, we have $E_2 = \{e_{21}, e_{22}, \dots, e_{2T_2}\}$ and $H_2 = \{h_{21}, h_{22}, \dots, h_{2T_2}\}$ for x_2.

We match every word embedding in E_1 with those in E_2, and match every context in H_1 with those in H_2. We also cross-match the contexts in H_1 (or H_2) with the words in E_2 (or E_1) to catch the matching patterns between contexts and words. Therefore, we obtain four interaction matrices $M_1, M_2, M_3, M_4 \in \mathbb{R}^{T_1 \times T_2}$:

$$M_1(i,j) = \tanh(h_{1i}^T h_{2j}),$$
$$M_2(i,j) = \tanh(\frac{1}{2}(\overrightarrow{h_{1i}}^T e_{2j} + \overleftarrow{h_{1i}}^T e_{2j})),$$
$$M_3(i,j) = \tanh(\frac{1}{2}(e_{1i}^T \overrightarrow{h_{2j}} + e_{1i}^T \overleftarrow{h_{2j}})) \; and$$
$$M_4(i,j) = \tanh(e_{1i}^T e_{2j}),$$

where we set $d_E = d_H$ to allow for the dot product between $\overrightarrow{h_{ij}}$ (or $\overleftarrow{h_{ij}}$) and e_{ij}. Then M_1, M_2, M_3, M_4 are stacked as a four-channel input for a CNN, whose output is passed through an MLP to predict the final matching degree y.

3 Experiments

Using three datasets, we show the superiority of DV-VAE over strong semi-supervised and supervised baselines.

3.1 Experimental Setup

Datasets. We experiment on three datasets: SNLI [Bowman *et al.*, 2015] for Natural Language Inference, Quora Question Pairs for Paraphrase Identification, and a Community Question Answering (CQA) dataset [Nakov *et al.*, 2015] for Question Answering. Statistics of these datasets are summarized in Table 2. (i) We perform simulated semi-supervised experiments with different amounts of labeled data along the same line as previous studies [Shen *et al.*, 2018; Zhao *et al.*, 2018]: for SNLI, we select 5.25%, 10.8% and 22.2% of the original train set to be $\mathbb{D}_l$ (i.e., approximately 28k, 59k and 120k labeled pairs), and remove the labels of the remaining data in the train set to make up $\mathbb{D}_u$; for Quora, we select 1k, 5k, 10k and 25k labeled pairs in the train set. We experiment on five random labeled/unlabeled splits

Models	SNLI			Quora			
	28k	59k	120k	1k	5k	10k	25k
LSTM-AE [Zhao *et al.*, 2018]	59.9	64.6	68.5	59.3	63.8	67.2	70.9
DeConv-AE [Shen *et al.*, 2018]	62.1	65.5	68.7	60.2	65.1	67.7	71.6
LSTM-ARAE [Zhao *et al.*, 2018]	62.5	66.8	70.9	-	-	-	-
LSTM-LVM [Shen *et al.*, 2018]	64.7	67.5	71.1	62.9	67.6	69.0	72.4
DeConv-LVM [Shen *et al.*, 2018]	**67.2**	**69.3**	**72.2**	**65.1**	**69.4**	**70.5**	**73.7**
Our interaction matcher*	70.88±0.84	73.70±0.70	77.35±0.16	63.68±1.44	71.97±0.60	72.39±1.64	75.92±0.58
DV-VAE	**71.19**±1.10	**74.54**±0.68	**78.79**±0.27	**68.12**±1.02	**73.07**±0.37	**74.69**±0.55	**77.04**±0.22

Table 1: Matching accuracy on SNLI and Quora, in percentage. *Our interaction matcher is trained on $\mathbb{D}_l$ in a supervised manner, while DV-VAE and the baselines are trained on $\mathbb{D}_l$ and $\mathbb{D}_u$ in a semi-supervised manner.

Datasets	#Train	#Dev	#Test	#Classes
SNLI	549,367	9,842	9,824	3
Quora	384,348	10,000	10,000	2
CQA	16,541	1,645	1,976	3

Table 2: Dataset statistics.

of the train set for each amount of labeled data, and report the mean and standard deviation of the matching accuracies. (ii) For the CQA dataset, the original train set is used as $\mathbb{D}_l$, and we additionally adopt WikiQA [Yang *et al.*, 2015], which has 29k QA pairs, as $\mathbb{D}_u$ by removing all its labels.[2]

Model Configurations. We set $d_E = 300$ and $d_Z = 500$. Word embeddings are initialized with Glove [Pennington *et al.*, 2014]. We share the parameters of Bi-LSTMs in the encoder and the interaction matcher. Hidden state size d_H is set to 300 for both directions. For the decoder, we choose a 3-layer dilated CNN, with dilation rates $[1, 2, 4]$. In all the bottleneck residual blocks, filter size is set to 3 and channel numbers are set to 300 internally and 600 externally. In the interaction matcher, we adopt a 2-layer CNN with filter sizes $5 \times 5 \times 8$ and $3 \times 3 \times 16$ such that a dynamic pooling is after the first layer to get 4×4 fixed-sized feature maps and a max pooling is after the second layer, followed by a 3-layer MLP with 16, 8 and C hidden units. ReLU is used as the nonlinearity and Batch Normalization is adopted in each layer.

Training Details. We use the reparameterization trick and sample one $\mathbf{z}$ from $q_\phi(\mathbf{z}|x)$ to estimate the variational lower bounds [Kingma and Welling, 2014]. α in Equ (5) is set to 20. We substitute the two KL-divergence terms in $\mathcal{L}(x_1, x_2, y)$ with $max(\gamma, \mathrm{KL}(q_\phi(\mathbf{z}_1|x_1)\|p(\mathbf{z}_1)) + \mathrm{KL}(q_\phi(\mathbf{z}_2|x_2)\|p(\mathbf{z}_2)))$ to force the decoder rely more on latent codes [Yang *et al.*, 2017]. We set $\gamma = 10$ for SNLI, and $\gamma = 20$ for the other experiments. SGD with momentum 0.9 and weight decay 1×10^{-3} is adopted in optimization. We use an initial learning rate of 3×10^{-3}. Batch size is tuned on $\{32, 64, 128\}$ for each experiment. We sample half of the minibatch from $\mathbb{D}_l$ and half from $\mathbb{D}_u$ in each iteration. We adopt early stopping where performance on dev set is evaluated every time $\mathbb{D}_l$ is traversed. A dropout rate of 0.1 is used in each layer of the decoder net. Experiments are implemented in PyTorch.

[2]We use the train/dev/test split of [Wang *et al.*, 2017] on Quora. Due to memory limitations, we truncate the texts in Quora, CQA and WikiQA to have no more than 100, 500 and 100 tokens, respectively.

3.2 Evaluations on Text Matching

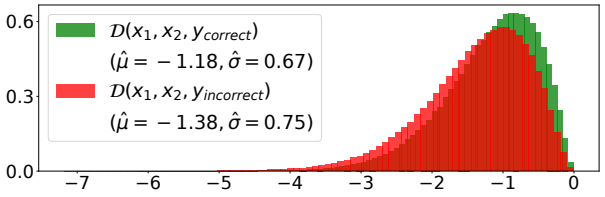
Natural Language Inference. First, we compare DV-VAE with semi-supervised baselines that combine autoencoders with sentence-encoding based matching models, and the results are reported in Table 1. Results indicate that DV-VAE consistently outperforms all the semi-supervised baselines by a large margin ($3.9\% \sim 6.5\%$) under all the 3 labeled data sizes. These results demonstrate the importance of incorporating the interaction view in DV-VAE for semi-supervised text matching. Second, we report in Table 1 the results from our interaction matcher trained on $\mathbb{D}_l$ in a supervised manner. DV-VAE consistently outperforms the supervised interaction matcher, verifying its effectiveness on using unlabeled data to improve supervised learning.

Paraphrase Identification. We get similar results on Quora, as shown in Table 1. DV-VAE's accuracy gains over the semi-supervised baselines are consistently more than 3% for all the 4 labeled data sizes. DV-VAE also achieves further accuracy gains over the supervised interaction matcher, and when labeled data is scarce ($|\mathbb{D}_l| = 1$k), the absolute improvement is up to 4.4%.

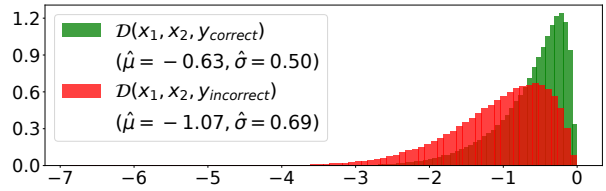
Community Question Answering. We compare our model with several strong supervised baselines in Table 3. These baselines and our interaction matcher are trained on $\mathbb{D}_l$ and DV-VAE is trained on $\mathbb{D}_l$ and $\mathbb{D}_u$ (WikiQA). Results show that DV-VAE outperforms all the baselines by leveraging the additional 29k unlabeled WikiQA sentence pairs, achieving an accuracy gain of 1.3% over the state of the art method KEHNN [Wu *et al.*, 2018]. Note that KEHNN leverages additional prior knowledge of the QA pairs while we leverage additional unlabeled QA pairs. This indicates that sufficient

Models	Accuracy
Attentive-LSTM	73.6
Match-LSTM	74.3
ARC-II	71.5
MatchPyramid	71.7
MV-LSTM	73.5
MultiGranCNN	74.3
KEHNN [Wu *et al.*, 2018] + prior knowledge	**75.3**
Our interaction matcher	74.4
DV-VAE + 29k unlabeled WikiQA data	**76.6**

Table 3: Matching accuracy on the CQA dataset, in percentage. Results in the first 7 rows are from [Wu et al., 2018].



(a) SNLI ($|\mathbb{D}_l| = 28$k)



(b) Quora ($|\mathbb{D}_l| = 25$k)

Figure 4: Distribution of the reward signal $\mathcal{D}(x_1, x_2, y)$ on $\mathbb{D}_u$.

amount of unlabeled data may play the role of prior knowledge in terms of the performance improvement.

3.3 Model Visualization

We first visualize the reward $\mathcal{D}(x_1, x_2, y)$ for the interaction matcher in Figure 4 to justify the implicit Co-Training mechanism. The distributions of $\mathcal{D}(x_1, x_2, y_{correct})$ and $\mathcal{D}(x_1, x_2, y_{incorrect})$ on $\mathbb{D}_u$ are distinguishable, and the mean for $\mathcal{D}(x_1, x_2, y_{correct})$ is larger than that for $\mathcal{D}(x_1, x_2, y_{incorrect})$, which is statistically significant ($p < 0.01$). This demonstrates that the interaction matcher may receive larger reward signals by predicting correct ys than incorrect ones on the unlabeled data. With the embedding matcher providing useful reward signals, the interaction matcher can effectively leverage unlabeled data.

We then visualize the learned decoder and embedding matcher in DV-VAE by generating labeled sentence pairs (x_1, x_2, y) from latent codes $\mathbf{z}_1$, $\mathbf{z}_2$ sampled from $p(\mathbf{z}) = \mathcal{N}(\mathbf{z}; \mathbf{0}, \mathbf{I})$. Some generated examples are shown in Table 4, demonstrating the capability of DV-VAE to learn the data manifold that is useful for semi-supervised classification.

x_1:	a child is playing with a soccer ball in the grass .
x_2:	a man is getting ready to throw a bowling ball down the lane .
y:	*neutral*
x_1:	the woman is sitting on the bench .
x_2:	a woman in a black shirt is walking down the street .
y:	*contradiction*
x_1:	a group of people are riding on a roller coaster .
x_2:	there is a group of people playing soccer .
y:	*contradiction*
x_1:	two dogs are playing with a red ball .
x_2:	the dog is running .
y:	*entailment*
x_1:	two young men , one wearing a white shirt and the other wearing a white shirt , are sitting on a bench .
x_2:	there is a man in a white shirt .
y:	*entailment*

Table 4: Sentence pairs generated by DV-VAE trained on SNLI ($|\mathbb{D}_l| = 28$k).

4 Related Work

Learning to match text sequences is a long standing problem and most state of the art methods use a compare-aggregate architecture [Wang and Jiang, 2017], such as DIIN [Gong *et al.*, 2018], CSRAN [Tay *et al.*, 2018], MwAN [Tan *et al.*, 2018] and KEHNN [Wu *et al.*, 2018]. [Lan and Xu, 2018] compared a broad range of text matching models over eight datasets. They all focus on supervised learning while we explore semi-supervised methods leveraging unlabeled text pairs to improve the performance of supervised methods.

Close to our work are recent applications of variational autoencoders [Kingma and Welling, 2014] and NVIL [Mnih and Gregor, 2014] in NLP. (i) Some focused on modeling a single piece of text: [Mnih and Gregor, 2014] and [Miao *et al.*, 2016] used bag of words methods for document modeling; [Bowman *et al.*, 2016b; Yang *et al.*, 2017] and others explored VAE and various improved models to generate natural language sentences; [Xu *et al.*, 2017] adopted semi-supervised VAE proposed in [Kingma *et al.*, 2014] for text classification. (ii) Others developed specific VAE structures for sequence transduction tasks such as sentence compression [Miao and Blunsom, 2016], machine translation [Zhang *et al.*, 2016], dialogue generation [Serban *et al.*, 2017]. (iii) To our knowledge, there are few studies modeling a pair of texts with VAE except [Shen *et al.*, 2018] that adopted deconvolutional networks in a VAE for semi-supervised text matching. However, this method matches texts from the embedding view only, while we further combine the interaction view.

Our work is also related to Multi-View Learning [Xu *et al.*, 2013] where features can be separated into distinct subsets (views). Particularly relevant are Co-Training [Blum and Mitchell, 1998] and Co-Regularization [Sindhwani *et al.*, 2005], where two models train each other on unlabeled data. However, instead of explicitly designing an algorithm or an objective to enable the Co-Training mechanism, we implicitly achieve it by maximizing the variational lower bound.

5 Conclusions

In this study, we have proposed Dual-View Variational AutoEncoder (DV-VAE) to unify the embedding view and the interaction view for semi-supervised text matching. Gradient analysis has also revealed an implicit Co-Training mechanism to explain the semi-supervised learning process. Finally, our experimental study has verified the effectiveness of DV-VAE. Further, our work is a step towards combining multi-view learning with neural network models, which seems a promising strategy for semi-supervised deep learning.

Acknowledgments

This work is supported in part by National Key R&D Program of China 2018YFB1700403, NSFC U1636210&61421003, Shenzhen Institute of Computing Sciences, and the Fundamental Research Funds for the Central Universities.

References

[Blum and Mitchell, 1998] Avrim Blum and Tom M. Mitchell. Combining labeled and unlabeled data with co-training. In *COLT*, pages 92–100, 1998.

[Bowman *et al.*, 2015] Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. A large annotated corpus for learning natural language inference. In *EMNLP*, pages 632–642, 2015.

[Bowman *et al.*, 2016a] Samuel R. Bowman, Jon Gauthier, Abhinav Rastogi, Raghav Gupta, Christopher D. Manning, and Christopher Potts. A fast unified model for parsing and sentence understanding. In *ACL*, pages 1466–1477, 2016.

[Bowman *et al.*, 2016b] Samuel R. Bowman, Luke Vilnis, Oriol Vinyals, Andrew M. Dai, Rafal Józefowicz, and Samy Bengio. Generating sentences from a continuous space. In *CoNLL*, pages 10–21, 2016.

[Dolan and Brockett, 2005] Bill Dolan and Chris Brockett. Automatically constructing a corpus of sentential paraphrases. In *IWP*, 2005.

[Gong *et al.*, 2018] Yichen Gong, Heng Luo, and Jian Zhang. Neural language inference over interaction space. In *ICLR*, 2018.

[He *et al.*, 2016] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

[Kingma and Welling, 2014] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. In *ICLR*, 2014.

[Kingma *et al.*, 2014] Diederik P. Kingma, Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling. Semi-supervised learning with deep generative models. In *NIPS*, pages 3581–3589, 2014.

[Lan and Xu, 2018] Wuwei Lan and Wei Xu. Neural network models for paraphrase identification, semantic textual similarity, natural language inference, and question answering. In *COLING*, pages 3890–3902, 2018.

[Miao and Blunsom, 2016] Yishu Miao and Phil Blunsom. Language as a latent variable: Discrete generative models for sentence compression. In *EMNLP*, 2016.

[Miao *et al.*, 2016] Yishu Miao, Lei Yu, and Phil Blunsom. Neural variational inference for text processing. In *ICML*, pages 1727–1736, 2016.

[Mnih and Gregor, 2014] Andriy Mnih and Karol Gregor. Neural variational inference and learning in belief networks. In *ICML*, pages 1791–1799, 2014.

[Mueller and Thyagarajan, 2016] Jonas Mueller and Aditya Thyagarajan. Siamese recurrent architectures for learning sentence similarity. In *AAAI*, pages 2786–2792, 2016.

[Nakov *et al.*, 2015] Preslav Nakov, Lluís Màrquez, Walid Magdy, Alessandro Moschitti, Jim Glass, and Bilal Randeree. Semeval-2015 task 3: Answer selection in community question answering. In *SemEval*, 2015.

[Pennington *et al.*, 2014] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *EMNLP*, 2014.

[Rocktäschel *et al.*, 2016] Tim Rocktäschel, Edward Grefenstette, Karl Moritz Hermann, Tomáš Kočiský, and Phil Blunsom. Reasoning about entailment with neural attention. In *ICLR*, 2016.

[Serban *et al.*, 2017] Iulian Vlad Serban, Alessandro Sordoni, Ryan Lowe, Laurent Charlin, Joelle Pineau, Aaron C. Courville, and Yoshua Bengio. A hierarchical latent variable encoder-decoder model for generating dialogues. In *AAAI*, pages 3295–3301, 2017.

[Shen *et al.*, 2018] Dinghan Shen, Yizhe Zhang, Ricardo Henao, Qinliang Su, and Lawrence Carin. Deconvolutional latent-variable model for text sequence matching. In *AAAI*, pages 5438–5445, 2018.

[Sindhwani *et al.*, 2005] Vikas Sindhwani, Partha Niyogi, and Mikhail Belkin. A co-regularization approach to semi-supervised learning with multiple views. In *ICML Workshop*, 2005.

[Tan *et al.*, 2018] Chuanqi Tan, Furu Wei, Wenhui Wang, Weifeng Lv, and Ming Zhou. Multiway attention networks for modeling sentence pairs. In *IJCAI*, 2018.

[Tay *et al.*, 2018] Yi Tay, Anh Tuan Luu, and Siu Cheung Hui. Co-stack residual affinity networks with multi-level attention refinement for matching text sequences. In *EMNLP*, pages 4492–4502, 2018.

[Wang and Jiang, 2017] Shuohang Wang and Jing Jiang. A compare-aggregate model for matching text sequences. In *ICLR*, 2017.

[Wang *et al.*, 2017] Zhiguo Wang, Wael Hamza, and Radu Florian. Bilateral multi-perspective matching for natural language sentences. In *IJCAI*, pages 4144–4150, 2017.

[Williams, 1992] Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8:229–256, 1992.

[Wu *et al.*, 2018] Yu Wu, Wei Wu, Can Xu, and Zhoujun Li. Knowledge enhanced hybrid neural network for text matching. In *AAAI*, pages 5586–5593, 2018.

[Xu *et al.*, 2013] Chang Xu, Dacheng Tao, and Chao Xu. A survey on multi-view learning. arXiv:1304.5634, 2013.

[Xu *et al.*, 2017] Weidi Xu, Haoze Sun, Chao Deng, and Ying Tan. Variational autoencoder for semi-supervised text classification. In *AAAI*, pages 3358–3364, 2017.

[Yang *et al.*, 2015] Yi Yang, Wen-tau Yih, and Christopher Meek. Wikiqa: A challenge dataset for open-domain question answering. In *EMNLP*, pages 2013–2018, 2015.

[Yang *et al.*, 2017] Zichao Yang, Zhiting Hu, Ruslan Salakhutdinov, and Taylor Berg-Kirkpatrick. Improved variational autoencoders for text modeling using dilated convolutions. In *ICML*, pages 3881–3890, 2017.

[Zhang *et al.*, 2016] Biao Zhang, Deyi Xiong, Jinsong Su, Hong Duan, and Min Zhang. Variational neural machine translation. In *EMNLP*, pages 521–530, 2016.

[Zhao *et al.*, 2018] Junbo Jake Zhao, Yoon Kim, Kelly Zhang, Alexander M. Rush, and Yann LeCun. Adversarially regularized autoencoders. In *ICML*, pages 5897–5906, 2018.