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) 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 different 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 Autoencoder

Suppose that we have a labeled sentence pair set $D_l$ and an unlabeled sentence pair set $D_u$. $(x_1, x_2, y) \in D_l$ 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, $(x_1, x_2) \in D_u$ denotes an unlabeled pair. Our goal is to develop a semi-supervised text matching model using both the labeled and unlabeled data $D_l$ and $D_u$, which can improve upon the performance of supervised text matching models using the labeled data $D_l$ 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 $z_1, z_2 \in \mathbb{R}^{D_z}$ are independently sampled from a prior $p(z)$, and are used to generate $x_1$ and $x_2$ through a decoder $p_0(\cdot|x)$. Latent codes $z_1, z_2$ are also fed into a sentence encoding-based matching model $p_0(y|z_1, z_2)$ to generate the matching degree $y$. The joint distribution can be explained by the following factorization:

$$p_0(x_1, x_2, y, z_1, z_2) = p(z_1)p_0(x_1|z_1)p(z_2)p_0(x_2|z_2)p_0(y|z_1, z_2),$$

where $p(z_1) = p(z_2) = p(z) = \mathcal{N}(z; 0, I)$ is a Gaussian prior. And $p_0(y|z_1, z_2)$ 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 $q_\phi(z_1, z_2, y|x_1, x_2)$ can be factorized as:

$$q_\phi(z_1, z_2, y|x_1, x_2) = q_\phi(z_1|x_1)q_\phi(z_2|x_2)q_\phi(y|z_1, z_2) = q_\phi(z_1|x_1)q_\phi(z_2|x_2)p_\theta(y|z_1, z_2),$$

where we model $q_\phi(y|z_1, z_2)$ by the embedding matcher $p_\theta(y|z_1, z_2)$. $q_\phi(z_1, z_2, y|x_1, x_2)$ can also be factorized as:

$$q_\phi(z_1, z_2, y|x_1, x_2) = q_\phi(y|x_1, x_2)q_\phi(z_1, z_2|x_1, x_2, y),$$

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: $q_\phi(y|x_1, x_2) = p_\theta(y|z_1, z_2)$. With this consistency assumption, we obtain the following from Equ (1) and Equ (2):

$$q_\phi(z_1, z_2|x_1, x_2, y) = q_\phi(z_1|x_1)q_\phi(z_2|x_2),$$

$$q_\phi(z_1, z_2, y|x_1, x_2) = q_\phi(y|x_1, x_2)q_\phi(z_1|x_1)q_\phi(z_2|x_2),$$

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

\[
\log p_\theta(x_1, x_2, y) \\
\geq E_{q_\phi(z_1, z_2|x_1, x_2, y)} \left[ \log \frac{p_\theta(x_1, x_2, y, z_1, z_2)}{q_\phi(z_1, z_2|x_1, x_2, y)} \right] \\
= E_{q_\phi(z_1|x_1)q_\phi(z_2|x_2)} \left[ \log p_\theta(x_1|z_1) + \log p_\theta(x_2|z_2) + \log p(y|z_1) + \log p(y|z_2) \right] \\
- \text{KL}(q_\phi(z_1|x_1)||\mu(z_1)) - \text{KL}(q_\phi(z_2|x_2)||\mu(z_2)) \\
= -\mathcal{L}(x_1, x_2, y).
\]

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

\[
\mathcal{L}(x_1, x_2, y) = -\mathcal{R}(x_1, x_2) - \mathcal{D}(x_1, x_2, y) \\
+ \text{KL}(q_\phi(z_1|x_1)||\mu(z_1)) + \text{KL}(q_\phi(z_2|x_2)||\mu(z_2)),
\]

(3)

where \(-\mathcal{R}(x_1, x_2) = E_{q_\phi(z_1|x_1)q_\phi(z_2|x_2)}[-\log p_\theta(x_1|z_1) - \log p_\theta(x_2|z_2)]\) is the reconstruction loss of \(x_1\) and \(x_2\); \(-\mathcal{D}(x_1, x_2, y) = E_{q_\phi(z_1|x_1)q_\phi(z_2|x_2)}[-\log p_\theta(y|z_1, z_2)]\) can be seen as an expected discriminative loss for the embedding matcher \(p_\theta(y|z_1, z_2)\); and the last two KL-divergence terms regularize the posteriors to be close to the priors.

For an unlabeled sentence pair \((x_1, x_2)\),

\[
\log p_\theta(x_1, x_2) \\
\geq E_{q_\phi(y|x_1, x_2, y|x_1, x_2)} \left[ \log \frac{p_\theta(x_1, x_2, y, z_1, z_2)}{q_\phi(z_1, z_2|x_1, x_2, y)} \right] \\
= E_{q_\phi(y|x_1, x_2)} \left[ E_{q_\phi(z_1, z_2|x_1, x_2, y)} \left[ \log \frac{p_\theta(x_1, x_2, y, z_1, z_2)}{q_\phi(z_1, z_2|x_1, x_2, y)} \right] \right] \\
- \log q_\phi(y|x_1, x_2) \\
= \sum_y q_\phi(y|x_1, x_2)(-\mathcal{L}(x_1, x_2, y) + \mathcal{H}[q_\phi(y|x_1, x_2)]) \\
\equiv -\mathcal{U}(x_1, x_2).
\]

(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 \(\alpha\):

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

(5)

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

\[
\mathcal{J} = \sum_{(x_1, x_2, y) \in \mathcal{D}_l} \mathcal{L}^\alpha(x_1, x_2, y) + \sum_{(x_1, x_2) \in \mathcal{D}_u} \mathcal{U}(x_1, x_2).
\]

(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 \(p_\theta(y|z_1, z_2)\) and the interaction matcher \(q_\phi(y|x_1, x_2)\) are simultaneously trained through the optimization of \(\mathcal{J}\) in Eqn (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 \(\phi_m\) and the parameters in \(p_\theta(y|z_1, z_2)\) as \(\theta_m\), 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 matches, 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 Eqn (4) w.r.t. \(\theta_m\) and \(\phi_m\), respectively.

For the embedding matcher \(p_\theta(y|z_1, z_2)\),

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

(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 \(q_{\phi_m}(y|x_1, x_2)\),

\[
\nabla_{\phi_m} \mathcal{U}(x_1, x_2) \\
= \sum_y \mathcal{L}(x_1, x_2, y) \nabla_{\phi_m} [q_{\phi_m}(y|x_1, x_2)] \\
- \nabla_{\phi_m} \mathcal{H}[q_{\phi_m}(y|x_1, x_2)] \\
- \nabla_{\phi_m} \mathcal{H}[q_{\phi_m}(y|x_1, x_2)] \\
= \sum_y q_{\phi_m}(y|x_1, x_2) \{\mathcal{D}(x_1, x_2, y) \nabla_{\phi_m} \}
- \log q_{\phi_m}(y|x_1, x_2) \} - \nabla_{\phi_m} \mathcal{H}[q_{\phi_m}(y|x_1, x_2)].
\]

(10)

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 matches 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 Eqn (7) becomes larger on correct \(ys,\) and the interaction matcher receives larger reward signals when it gives correct predictions in Eqn (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 \(q_\phi(z|x)\), a decoder \(p_\theta(x|z)\), an embedding matcher \(p_\theta(y|z_1, z_2)\) and an interaction matcher \(q_\phi(y|x_1, x_2)\).

---

To derive Eqn (8), note that \(\phi_m\) does not exist in \(\mathcal{L}(x_1, x_2, y)\); to derive Eqn (9), note that \(\sum_y K \nabla_{\omega} q(y; \omega) = K \nabla_{\omega} \sum_y q(y; \omega) = K \nabla_{\omega} 1 = 0\) 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 Eqn (10), use \(\nabla_{\omega} q(y; \omega) = q(y; \omega) \nabla_{\omega} \log q(y; \omega).\)
Encoder $q_{θ}(z|x)$. 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 $q_{θ}(z|x) = \mathcal{N}(z; μ, \text{diag}(σ^2))$.

Decoder $p_{θ}(x|z)$. 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 $z$ 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 a masked dilated 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 $\{2^0, 2^1, 2^2, \ldots\}$, 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 $p_{θ}(w_j|w_{<j}, z)$ for all time steps $j \in [1, \ldots, T]$. Then the reconstruction probability of text sequence $x$ is computed as $p_{θ}(x|z) = \prod_{j=1}^{T} p_{θ}(w_j|w_{<j}, z)$.

Embedding matcher $p_{θ}(y|z_1, z_2)$. We adopt an MLP taking as input the concatenation of $z_1, z_2$, the element-wise difference $z_1 - z_2$ and the element-wise product $z_1 \odot z_2$:

$$p_{θ}(y|z_2, z_2) = \text{softmax}(\text{MLP}([z_1; z_2; z_1 - z_2; z_1 \odot z_2]))$$

Interaction matcher $q_{θ}(x_1, x_2)$. For $x_1 = \{w_1, w_2, \ldots, w_{T_1}\}$, we denote the word embedding sequence as $E_1 = \{e_1, e_2, \ldots, e_{T_1}\}$, where $e_{ij} \in \mathbb{R}^{d_E}$ is the word embedding for token $w_{ij}$. Then a bidirectional LSTM (Bi-LSTM) is adopted to get a context sequence $H_1 = \{h_{11}, h_{12}, \ldots, h_{1T_1}\}$, where $h_{ij} \leftarrow [h_{ij}; h_{ij}]$ is the concatenation of the corresponding forward and backward hidden states of the Bi-LSTM, and $h_{2j}$, $h_{ij} \in \mathbb{R}^{d_H}$. Similarly, we have $E_2 = \{e_2, e_2, \ldots, e_{2T_2}\}$ and $H_2 = \{h_{21}, h_{22}, \ldots, h_{2T_2}\}$ 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 $M_1, M_2, M_3, M_4 \in \mathbb{R}^{T_1 \times T_2}$:

$$M_1(i, j) = \text{tanh}(h_{i1}^T h_{j2}),$$

$$M_2(i, j) = \frac{1}{2} (h_{i1}^T e_{j2} + h_{i2}^T e_{j2}),$$

$$M_3(i, j) = \frac{1}{2} (e_{i1}^T h_{j2} + e_{i2}^T h_{j2})$$

$$M_4(i, j) = \text{tanh}(e_{i1}^T e_{j2}),$$

where we set $d_E = d_H$ to allow for the dot product between $h_{ij}$ (or $h_{ij}$) 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 $D_t$ (i.e., approximately 28k, 59k and 120k labeled pairs), and remove the labels of the remaining data in the train set to make up $D_u$; for Quora, we select 1k, 5k, 10k and 25k labeled pairs in the train set. We experiment on five random labeled/unlabeled splits.
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 \(\mathcal{D}_l\), and we additionally adopt WikiQA [Yang et al., 2015], which has 29k QA pairs, as \(\mathcal{D}_u\) by removing all its labels.2

**Model Configurations.** We set \(d_E = 300\) and \(d_F = 500\). 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 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 \(4 \times 4\) 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 \(z\) from \(q_\phi(z|x)\) to estimate the variational lower bounds [Kingma and Welling, 2014]. \(\alpha\) in Equ (5) is set to 20. We substitute the two KL-divergence terms in \(\mathcal{L}(x_1, x_2, y)\) with \(\max(\gamma, \text{KL}(q_\psi(z_1|x_1)||p(z_1)) + \text{KL}(q_\psi(z_2|x_2)||p(z_2)))\) 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 \(1 \times 10^{-3}\) is adopted in optimization. We use an initial learning rate of \(3 \times 10^{-3}\). Batch size is tuned on \([32, 64, 128]\) for each experiment. We sample half of the minibatch from \(\mathcal{D}_l\) and half from \(\mathcal{D}_u\) in each iteration. We adopt early stopping where performance on dev set is evaluated every time \(\mathcal{D}_l\) is traversed. A dropout rate of 0.1 is used in each layer of the decoder net. Experiments are implemented in PyTorch.

3We 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% ~ 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 \(\mathcal{D}_l\) 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 (\(|\mathcal{D}_l| = 1k\), 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 \(\mathcal{D}_l\) and DV-VAE is trained on \(\mathcal{D}_l\) and \(\mathcal{D}_u\) (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

<table>
<thead>
<tr>
<th>Models</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attentive-LSTM</td>
<td>73.6</td>
</tr>
<tr>
<td>Match-LSTM</td>
<td>74.3</td>
</tr>
<tr>
<td>ARC-II</td>
<td>71.5</td>
</tr>
<tr>
<td>MatchPyramid</td>
<td>71.7</td>
</tr>
<tr>
<td>MV-LSTM</td>
<td>73.5</td>
</tr>
<tr>
<td>MultiGranCNN</td>
<td>74.3</td>
</tr>
<tr>
<td>KEHNN [Wu et al., 2018]</td>
<td>75.3</td>
</tr>
<tr>
<td>Our interaction matcher</td>
<td>74.4</td>
</tr>
<tr>
<td>DV-VAE + 29k unlabeled WikiQA data</td>
<td>76.6</td>
</tr>
</tbody>
</table>

Table 3: Matching accuracy on the CQA dataset, in percentage. Results in the first 7 rows are from [Wu et al., 2018].
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 $D(x_1, x_2, y)$ for the interaction matcher in Figure 4 to justify the implicit Co-Training mechanism. The distributions of $D(x_1, x_2, y_{\text{correct}})$ and $D(x_1, x_2, y_{\text{incorrect}})$ on $\mathbb{D}_u$ are distinguishable, and the mean for $D(x_1, x_2, y_{\text{correct}})$ is larger than that for $D(x_1, x_2, y_{\text{incorrect}})$, which is statistically significant ($p < 0.01$). This demonstrates that the interaction matcher may receive larger reward signals by predicting correct $y$s 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 $z_1, z_2$ sampled from $p(z) = \mathcal{N}(z; 0, 1)$. 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.

### 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


