A Deep Decomposable Model for Disentangling Syntax and Semantics in Sentence Representation

Dingcheng Li, Hongliang Fei, Shaogang Ren, Ping Li
Cognitive Computing Lab (CCL)
Baidu Research USA
10900 NE 8th St. Bellevue, WA 98004, USA
{lidingcheng,hongliangfei,shaogangren,liping11}@baidu.com

Abstract

Recently, disentanglement based on a generative adversarial network or a variational autoencoder has significantly advanced the performance of diverse applications in CV and NLP domains. Nevertheless, those models still work on coarse levels in the disentanglement of closely related properties, such as syntax and semantics in human languages. This paper introduces a deep decomposable model based on VAE to disentangle syntax and semantics by using total correlation penalties on KL divergences. Notably, we decompose the KL divergence term of the original VAE so that the generated latent variables can be separated in a more clear-cut and interpretable way. Experiments on benchmark datasets show that our proposed model can significantly improve the disentanglement quality between syntactic and semantic representations for semantic similarity tasks and syntactic similarity tasks.

1 Introduction

Recently, disentangled representations have significantly advanced the performance of several applications in NLP. For example, disentanglement has been used to separating representation of attributes such as sentiment from contents (Fu et al., 2018; John et al., 2019), understanding subtleties in component modeling (Esmaeili et al., 2019), detecting anomalies (Hou et al., 2021), and learning sentence representations that split the syntax and the semantics (Ju et al., 2021). They are also used to boost text generation (Iyyer et al., 2018; Jain et al., 2018) or calculating the semantic or syntactic similarity between sentences (Chen et al., 2018).

In this paper, we focus on the task of separating syntax and semantics in sentence representation learning. Unlike previous supervised approaches that usually resort to syntactic parsers to handle syntax processing, our approach separates syntactic and semantic variables by disentangling hidden states of deep neural nets in a self-learning and unsupervised fashion.

The first work focusing on the separation of syntax and semantics from hidden variables is Chen et al. (2019). They proposed a deep generative model based on VAE with two latent variables to represent syntax and semantics. The generative model comprises von Mises Fisher (vMF) and Gaussian priors on the semantic and syntactic latent variables, and a deep BOW decoder conditioning on these latent variables. Following previous work, they train this model by optimizing the Evidence Lower Bound (ELBO) with a VAE-like (Kingma and Welling, 2014) objective.

However, their approach still generates a rough decomposition and thus may fail to disentangle syntax and semantics at a finer granularity. To address this weakness, we propose a decomposable variational autoencoder (DecVAE) to allow hidden variables factorizable. From a modeling perspective, factorizable representations with statistically independent variables usually obtained in an unsupervised or semi-supervised manner can distill information into a compact form, which is semantically useful for downstream tasks. From an application perspective, different words or phrases in sentences represent various entities with variant roles. It is necessary to utilize decomposable latent variables to capture a variety of entities with different semantic meanings.

Towards building a finer-grained disentanglement, motivated by FactorVAE (Kim and Mnih, 2018), we extend the work in Chen et al. (2019) and use total correlation (Watanabe, 1960) (TC) as a penalty term to obtain a deeper and meaningful factorization of syntactic and semantic latent variables. To make TC more discriminative, we also integrate multi-head attention into this framework. DecVAE can identify and cluster hierarchically independent semantic components in natural language text, which exhibits hierarchical linguistic

Findings of the Association for Computational Linguistics: EMNLP 2021, pages 4300–4310

structure (Sanh et al., 2019), and the corresponding syntax and semantics interact with each other. For experiments, we evaluate learned semantic representations on the SemEval semantic textual similarity (STS) tasks. Following the protocol in Chen et al. (2019), we predict the syntactic structure of an unseen sentence to be the one similar to its nearest neighbor, determined by the latent syntactic representation in a large dataset of annotated sentences. Experiments show that DecVAE achieves the best performance on all tasks when learned representations are mostly disentangled.

Contributions. Firstly, we propose a generic DecVAE to disentangle semantics and syntax based on the total correlation of KL divergence. Secondly, DecVAE is also integrated with a multi-head attention network to cluster embedding vectors so that corresponding word embeddings are more discriminative. Thirdly, results after integrating DecVAE in disentangling syntax from semantics achieve SOTA performances, confirming DecVAE's effectiveness.

2 Background and Related Work

2.1 VAEs for Disentanglement

The variational autoencoder (VAE) (Kingma and Welling, 2014) is a latent variable model that pairs a top-down generator with a bottom-up inference network. Different from traditional maximum-likelihood estimation (MLE) approach, VAE training is done by *evidence lower bound* (ELBO) optimization in order to overcome the intractability of posterior. Basically, the objective function of VAE is represented as:

$$\mathbb{E}_{\mathbf{z}\sim q(\mathbf{Z}|\mathbf{X})}[\log p(\mathbf{X}|\mathbf{Z})] - \beta\mathbf{KL}(q(\mathbf{Z}|\mathbf{X})||p(\mathbf{Z}))$$

When $\beta = 1$, this is the standard VAE. When $\beta > 1$, it becomes β-VAE (Higgins et al., 2017), which attempts to learn a disentangled representation by optimizing a heavily penalized objective.

Vanilla VAEs cannot disentangle latent variables. PixelGAN Autoencoders (Makhzani and Frey, 2017) further break down the KL term as:

$$\mathbf{KL}(q(\mathbf{Z}|\mathbf{X})||p(\mathbf{Z})) = I(\mathbf{X};\mathbf{Z}) + \mathbf{KL}(q(\mathbf{Z})||p(\mathbf{Z})) \quad (1)$$

where $I(x;z)$ is the mutual information under the joint distribution $p(x)q(z|x)$. Penalizing the $KL(q(z)||p(z))$ term pushes $q(z)$ towards the factorial prior $p(z)$, encouraging independence in the dimensions of z and thus disentangling.

Alternatively, FactorVAE approaches this problem with total correlation penalty (Kim and Mnih, 2018), which we adopt for our work. FactorVAE achieves similar disentangling results while preserving good quality of reconstruction by augmenting the vanilla VAE objective with a term directly encouraging independence in the code distribution:

$$\mathbb{E}_{\mathbf{z}\sim q(\mathbf{Z}|\mathbf{X})}[\log p(\mathbf{X}|\mathbf{Z})] - \mathbf{KL}(q(\mathbf{Z}|\mathbf{X})||p(\mathbf{Z})) - \gamma\mathbf{KL}(q(\mathbf{Z})||\bar{q}(\mathbf{Z}))$$

where $\bar{q}(\mathbf{z}) := \prod_{j=1}^{K} q(z_j)$. The FactorVAE's objective is also a lower bound on the marginal log likelihood $\mathbb{E}_p[\log p(\mathbf{X})]$. $\mathbf{KL}(q(\mathbf{Z})||\bar{q}(\mathbf{Z}))$ is known as "Total Correlation" (TC) (Watanabe, 1960), a popular measure of dependence for multiple random variables.

2.2 Disentanglement in NLP

Disentanglement in NLP has strong connections with LDA (Blei et al., 2003; Blei and Lafferty, 2006). In particular, neural topic models, that use belief networks (Mnih and Gregor, 2014; Li et al., 2019b) or enforce the Dirichlet prior via Gaussian or Wassertein autoencoders (Nan et al., 2019; Li et al., 2018), associate topic learning to disentanglement with component analysis. Later on, seq2seq VAE represent disentangled topics via continuous representations (Dieng et al., 2017; Ding et al., 2018; Bowman et al., 2016; Yang et al., 2017). Srivastava and Sutton (2017) combines LDA and VAE for topic detection and Pergola et al. (2021) proposes to consider latent topics as generative factors to be disentangled to improve discriminative power of topics.

Meanwhile, a growing amount of work start to explore neural learning disentangled/component representations to diverse NLP tasks. For example, we see such applications in sentiment analysis and style transfer (Hu et al., 2017; Li et al., 2019a), morphological reinflectioon (Zhou and Neubig, 2017), semantic parsing (Yin et al., 2018), text generation (Wiseman et al., 2018), sequential labeling (Chen et al., 2018), text-based variational autoencoder (Miao et al., 2016), etc.

Although much work has been done on grammatical and semantic analysis, there are few explorations on disentangling syntax and semantics. The disentanglement between syntax and semantics is quite challenging since they are heavily entangled. Except under some circumstances where there are no ambiguities, such as some unique proper names,

it is usually difficult to find absolute borderlines among words, phrases, or entities.

The work of VGVAE (Chen et al., 2019) is the latest one quite relevant to our work, wherein they assume that a sentence is generated by conditioning on two independent latent variables: semantic variable $\mathbf{z}_{sem}$ and syntactic variable $\mathbf{z}_{syn}$. For inference, they assume a factored posterior is produced and a lower bound on marginal log-likelihood is maximized in the generative process. The corresponding inference and generative models are two independent word averaging encoders with additional linear feed-forward neural networks and a feed-forward neural network with the output being a bag of words or an RNN.

Compared with their work, we aim to construct a more generic work by deploying the decomposability of KL divergence, thus discovering more subtle components from latent variables. Consequently, the VAE framework can do better disentanglement with more fine-grained decomposed parts. Further, we can flexibly add regularities to guide the decomposition to generate more interpretable and controllable elements from decoders.

3 Proposed Approach

In this work, we are developing a generative model named Decomposable VAE (DecVAE). Although our proposed approach is applicable to any disentangled tasks in NLP, we focus on disentangling semantic and syntactic information from sentence representations. We extend VGVAE model (Chen et al., 2019) to incorporate the total correlation as a penalty term to enable latent variable factorization.

3.1 Decomposable VAE

Our model is essentially based on VAE, namely, composed of a term of computing loglikelihood of input data given latent variables, and terms of computing KL divergences between posterior variational probabilities of hidden variables given input data and the prior probabilities of hidden variables.

Let $x_1, ..., x_T$ be a sequence of T tokens (words), conditioned on a continuous latent variable $\mathbf{z}$. As a usual practice, for example, like the assumption in Latent Dirichlet Allocations (LDA) (Blei et al., 2003), we have a conditional independence assumption of words on $\mathbf{z}$:

$$p_\theta(x_1, ..., x_T) = \int \prod_{t=1}^{T} p_\theta(x_t|\mathbf{z}) p_\theta(\mathbf{z}) d\mathbf{z}$$

Model parameters θ can be learned via the variational lower-bound (Kingma and Welling, 2014)

$$\mathcal{L}(\theta, \phi; \mathbf{X}) \geq \frac{1}{T} \sum_{t=1}^{T} (\mathbb{E}_{\mathbf{z}\sim q_\phi}[\log p_\theta(x_t|\mathbf{z})] - \mathbf{KL}(q_\phi(\mathbf{z}|x_t)||p_\theta(\mathbf{z}))) \quad (2)$$

where $q_\phi(\mathbf{z}|x_t)$ is the encoder (recognition model or inference model), parameterized by ϕ, i.e., the approximation to true posterior $p_\theta(\mathbf{z}[x_t)$. The distribution $p_\theta(\mathbf{z})$ is the prior for $\mathbf{z}$.

As studied in Sanh et al. (2019), natural languages can be regarded as a manifold, since it is hierarchically organized, and the corresponding syntax and the semantics interact in an intricate space. Based on the observation that different words or phrases in sentences represent different entities with different roles, either grammatical or semantic, and potentially interact with each other, we guide the generations of latent variables in the VAE corresponding to entities in sentences by designing a VAE with decomposable latent variables. Hence our proposed DecVAE can identify hierarchically independent components from natural languages. Furthermore, the reconstruction network may generate words or phrases sequentially.

DecVAE will learn a decoder that maps the latent space $\mathcal{Z}$ (learned by the encoder from input samples) to this language manifold $\mathcal{X}$. Let $\mathbf{Z} = [\mathbf{z}^1, \cdots, \mathbf{z}^K] \in \mathcal{Z}$ be the latent variable of the decoder and $\mathbf{z}^k$ to represent the k-th component of the latent variables. In addition, we also add a $\mathbf{z}_0$ to each $\mathbf{z}^k$, a special latent variable to encodes the overall properties of the generated sentences and the correlations between different grammatical and semantic components. Let $(\bar{\mathbf{x}}, \bar{\mathbf{f}}) = [(\bar{\mathbf{x}}^1, \bar{\mathbf{f}}^1), \cdots, (\bar{\mathbf{x}}^K, \bar{\mathbf{f}}^K)]$ be the variables for the output of the decoder (each element is a tuple composed of the generated token index in the vocabulary and its component index), where $\mathbf{z}^k$ controls the properties of k-th component $\bar{\mathbf{x}}^k$.

Firstly, we assume that the components are conditionally independent with each other given the latent variables, i.e.,

$$(\bar{\mathbf{x}}^i, \bar{\mathbf{f}}^i) \perp (\bar{\mathbf{x}}^j, \bar{\mathbf{f}}^j)|\mathbf{Z}, \text{if } i \neq j.$$

We also have the following independent assumption about the components and latent variables,

$$(\bar{\mathbf{x}}^i, \bar{\mathbf{f}}^i) \perp \mathbf{z}^j|\mathbf{z}_0^j, \text{if } i \neq j. \quad (3)$$

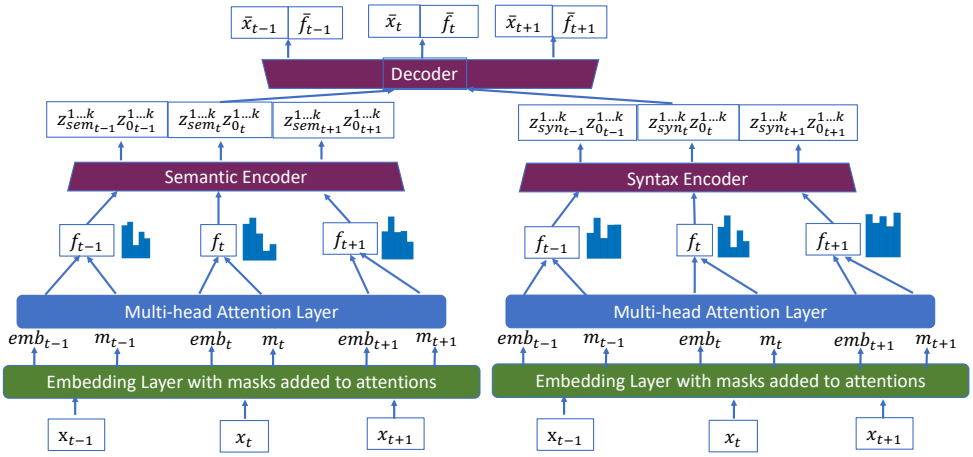

Figure 1: The proposed model consists of four layers. From bottom to top, they are embedding layer, multi-head attention layer, encoder, and decoder. Different from the usual network structure, the first three layers comprise three parallel independent layers, one for semantic and one for syntax. The attention layers yield K-dim attention weights $\mathbf{f}$, so that ensemble of K weighted embeddings are working in both semantic and syntax encoders.

Let $\bar{\mathbf{y}} = (\bar{\mathbf{x}}, \bar{\mathbf{f}})$ and each $\bar{\mathbf{y}}^k = (\bar{\mathbf{x}}^k, \bar{\mathbf{f}}^k)$. We have the following distributions for generated tokens:

$$p(\bar{\mathbf{y}}|\mathbf{z}) = p(\bar{\mathbf{y}}^1, \cdots, \bar{\mathbf{y}}^K|\mathbf{z}_0, \mathbf{z}^1, \cdots, \mathbf{z}^K)$$
$$= \prod_{k=1}^{K} p(\bar{\mathbf{y}}^k|\mathbf{z}_0^k, \mathbf{z}^1, \cdots, \mathbf{z}^K) = \prod_{k=1}^{K} p(\bar{\mathbf{y}}^k|\mathbf{z}_0^k, \mathbf{z}^k)$$

This model attempts to encode each component's individual features (tokens, words, or phrases) and the global latent factors for the sentence.

3.2 Objective Function

We propose to decompose the two terms of calculating KL divergence following Eq. (1). Meanwhile, along the thread of our proposed DecVAE, we add the global controller variable $\mathbf{z}_0$. This design shares some similarities with the component segmentation in computer vision, such as MONet (Burgess et al., 2019). MONet shows that an attention network layer improves component segmentation as well as component disentanglement, in which a variable, f, the representation of the attention, is deployed there. Taking these into consideration, our model is defined as following. Let $\mathbf{z}_{syn} = [\mathbf{z}_{syn}^1, \cdots, \mathbf{z}_{syn}^K]$ be the syntactic latent variable, we define an equation for syntax based on the decomposable nature of latent variables as:

$$\mathbf{KL}(q_\phi(\mathbf{z}_{syn}^k|\mathbf{x})||p_\theta(\mathbf{z}_{syn}^k)) = I_{q_\phi}(\mathbf{x}, \mathbf{f}^k; \mathbf{z}_{syn}^k, \mathbf{z}_0^k)$$
$$+ \sum_{i,j}[\mathbf{KL}(q(\mathbf{z}_{syn}^{k_i}, \mathbf{z}_0^{k_j})||p(\mathbf{z}_{syn}^{k_i}, \mathbf{z}_0^{k_j})) \quad (4)$$
$$+ \beta\mathbf{KL}(q_\phi(\mathbf{z}_{syn}^k, \mathbf{z}_0^k)||\prod_i q_\phi(\mathbf{z}_{syn}^{k_i})\prod_j q_\phi(\mathbf{z}_0^{k_j}))]$$

and a similar equation for semantics as

$$\mathbf{KL}(q_\phi(\mathbf{z}_{sem}^k|\mathbf{x})||p_\theta(\mathbf{z}_{sem}^k)) = I_{q_\phi}(\mathbf{x}, \mathbf{f}^k; \mathbf{z}_{sem}^k, \mathbf{z}_0^k)$$
$$+ \sum_{i,j}\mathbf{KL}(q(\mathbf{z}_{sem}^{k_i}, \mathbf{z}_0^{k_j})||p(\mathbf{z}_{sem}^{k_i}, \mathbf{z}_0^{k_j})) \quad (5)$$
$$+ \beta\mathbf{KL}(q_\phi(\mathbf{z}_{sem}^k, \mathbf{z}_0^k)||\prod_i q_\phi(\mathbf{z}_{sem}^{k_i})\prod_j q_\phi(\mathbf{z}_0^{k_j}))],$$

where i, j refer to indices of tokens and $\mathbf{z}_*^{k_i}, * \in \{sem, syn, 0\}$ indicates the latent variable value at the i-th token. In Eq. (4) and Eq. (5), the second and third terms are derived from minimization of total correlations as in Esmaeili et al. (2019); Jeong and Song (2019). The second term decomposes each hidden vector of syntax and semantics into smaller categories in a hierarchical fashion so that we can have more subtle disentanglements of each syntactic or semantic components.

The third term in Eq. (4) and Eq. (5) is derived from the standard equation of total correlation,

$$TC(\mathbf{z}^\mathbf{k}) = \mathbb{E}\Big[\log(\frac{q_\phi(\mathbf{z}^\mathbf{k})}{\prod_i q_\phi(\mathbf{z}_i^k)}\Big] = KL(q_\phi(\mathbf{z}^k)||\prod_d q_\phi(\mathbf{z}_i^k)))$$

Namely, we deploy this technique to penalize the total correlation (TC) for enforcing disentanglement of the latent factors. To compute the second term, we use the weighted version for estimating the distribution value of $q(\mathbf{z})$.

3.3 The Network Structure

With the above derivations as our basis, we construct our network structure as shown in Figure 1. From bottom to top, the input sentences are con-

verted to embedding vectors. Meanwhile, there is a mask input with each mask m_k showing whether each word or phrase x_t appears in each sentence. Outputs from this layer are fed to a multi-head attention layer to generate attention weights f_t. Following-up is the dot product between the embedding of x_t and its attention weight f_t.

Since we are modeling both semantics and syntax of input sentences, the attention procedure is processed twice with different initialization. The results are passed into the semantic encoder and syntax encoder, respectively. Each encoder yields their hidden variables, ($\mathbf{z}_{sem_t}^{1\cdots k}$, $\mathbf{z}_{0_t}^{1\cdots k}$) and ($\mathbf{z}_{syn_t}^{1\cdots k}$, $\mathbf{z}_{0_t}^{1\cdots k}$). A similar idea is implemented in recent work from computer vision domain (CV), MONet (Burgess et al., 2019). Differently, in their work, f_k is generated sequentially with an attention network while we generate attention all at once with multi-head attention, which is proven successful in the transformer model (Vaswani et al., 2017).

To incorporate recurrent neural networks for decoding, we take a similar structure described in SNAIL (Mishra et al., 2018). Namely, the self-attention mechanism from the transformer is combined with a temporal convolution. Next, the element-wise multiplication of embedding vector and focus masks generate hidden vectors, which are fed into semantic encoder and syntax encoder respectively to be encoded as a pair of variables $(\mathbf{z}^k, \mathbf{z}_0^k)$. The two groups of hidden component vectors are concatenated into the decoder. We obtain the reconstructed words/phrases $\bar{\mathbf{x}}$, and their component distribution $\bar{\mathbf{f}}^k$, similar to a component assignment and consistent to the weights $\mathbf{f}^k$.

3.4 Multi-task Training and Inference

With the product of embedding vector $\mathbf{emb}_t$ and their corresponding focus mask $\mathbf{m}_t$ as the encoder's input, $(\mathbf{z}^k, \mathbf{z}_0^k)$ as the latent variable and $(\bar{\mathbf{x}}, \bar{\mathbf{m}}^k)$ as the output of the decoder, the loss for component k is given by

$$
\begin{aligned}
&\Psi_k(\mathbf{x}, \mathbf{f}^k; \theta, \phi, a, e, d) \qquad (6)\\
&= -\mathbb{E}_{q_\phi^e(\mathbf{z}^k, \mathbf{z}_0^k|\mathbf{x}, \mathbf{f}^k)}\left[\mathbf{f}^k \log p_\theta^d(\mathbf{x}|\mathbf{z}^k, \mathbf{z}_0^k)\right]\\
&\quad + \mathrm{KL}\big(q_\phi^e(\mathbf{z}^k, \mathbf{z}_0^k|\mathbf{x}, \mathbf{f}^k)||p(\mathbf{z}^k, \mathbf{z}_0^k)\big)\\
&\quad + \gamma\mathbf{KL}(q_\phi^a(\mathbf{f}^k|\mathbf{x})||p_\theta^d(\bar{\mathbf{f}}^k|\mathbf{z}^k, \mathbf{z}_0^k))
\end{aligned}
$$

Here a, e and d refer to multi-head attention layer, encoder and decoder layer respectively, θ and ϕ are parameters for the likelihood and variational distribution respectively, the local hidden variable

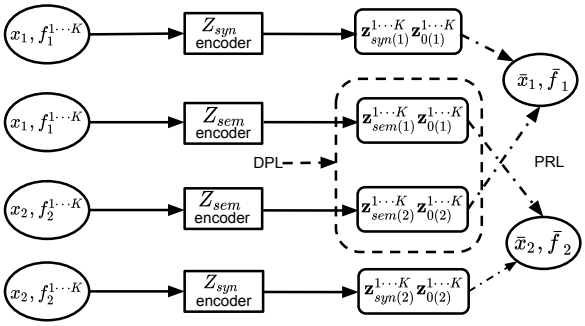

Figure 2: Diagram of the training process for auxiliary losses: discriminative paraphrase loss (DPL; dashed lines) and paraphrase reconstruction loss (PRL; dash-dotted lines). Different from Chen et al. (2019), each input of encoders consists of embeddings of the sentence $\mathbf{x}_t$ and their component distributions, $\mathbf{f}_t^{1\cdots k}$. Each output of encoders consists of hidden variables $\mathbf{z}_{sem_t}^{1\cdots k}$ and $\mathbf{z}_{0_t}^{1\cdots k}$. Each output of decoders consists of predicted embeddings of each sentence $\bar{x}_t$ and their predicted component distributions, $\bar{\mathbf{f}}_t$.

$\mathbf{z}^k = [\mathbf{z}_{sem}^k, \mathbf{z}_{syn}^k]$ and the global hidden variable $\mathbf{z}_0^k = [\mathbf{z}_{sem(0)}^k, \mathbf{z}_{syn(0)}^k]$ and $\gamma \geq 0$ is a hyper-parameter. The overall loss is

$$\mathcal{L}_{\text{VAE}}(\mathbf{x}; a, e, d) = \sum_{k=1}^{K} \Psi_k(\mathbf{x}, \mathbf{f}^k; \theta, \phi, a, e, d).$$

Loss Function Components. As seen from Eq. (6), our loss function is composed of three parts, which can be realized by our objective functions described in Eq. (4) and Eq. (5). Furthermore, following the success of multi-task training in Chen et al. (2019), we introduce three auxiliary objectives: paraphrase reconstruction loss (PRL), discriminative paraphrase loss (DPL) and word position loss (WPL). The purpose is to encourage $\mathbf{z}_{sym}$ to better capture semantic information and $\mathbf{z}_{syn}$ to better capture syntactic information.

Paraphrase Reconstruction Loss Function. As shown in Figure 2, we swap the semantic variables, keep the syntactic variables and attempt to reconstruct the sentences. We model sentences with paraphrase relationships $\mathbf{x}_1$ and $\mathbf{x}_2$ to be generated with the same semantic latent variables. The basic assumption is still that semantic information is equivalent between a paraphrase pair. But differently, our PRL involve more variables, including the common latent factor $\mathbf{z}_0$ and the focus mask variables $\mathbf{f}_k$. Therefore, our PRL is defined as,

$$\mathop{\mathbb{E}}_{\substack{\mathbf{z}_{sem(2)}\sim q_\phi^e(\mathbf{1})\\ \mathbf{z}_{syn(1)}\sim q_\phi^e(\mathbf{2})}}[-\log p_\theta^d(\bar{\mathbf{x}}_1|(\mathbf{z}_{sem(2)},\mathbf{z}_{0(2)}),(\mathbf{z}_{syn(1)},\mathbf{z}_{0(1)})]+$$

$$\mathop{\mathbb{E}}_{\substack{\mathbf{z}_{sem(1)}\sim q_\phi^e(\mathbf{3})\\ \mathbf{z}_{syn(2)}\sim q_\phi^e(\mathbf{4})}}[-\log p_\theta^d(\bar{\mathbf{x}}_2|(\mathbf{z}_{sem(1)},\mathbf{z}_{0(1)}),(\mathbf{z}_{syn(2)},\mathbf{z}_{0(2)})]$$

where

$$q_\phi^e(\mathbf{1}) = q_\phi^e((\mathbf{z},\mathbf{z}_0)_{sem}|\bar{\mathbf{x}}_2,\bar{\mathbf{f}}_2), q_\phi^e(\mathbf{2}) = q_\phi^e((\mathbf{z},\mathbf{z}_0)_{syn}|\bar{\mathbf{x}}_1,\bar{\mathbf{f}}_1),$$
$$q_\phi^e(\mathbf{3}) = q_\phi^e((\mathbf{z},\mathbf{z}_0)_{sem}|\bar{\mathbf{x}}_1,\bar{\mathbf{f}}_1), q_\phi^e(\mathbf{4}) = q_\phi^e(\mathbf{z},\mathbf{z}_0)_{syn}|\bar{\mathbf{x}}_2,\bar{\mathbf{f}}_2).$$

Discriminative Paraphrase Loss. The Discriminative Paraphrase Loss (DPL) attempts to learn to encourage sentences with paraphrase relationships to have higher similarities while those without such relationships to have lower similarities. Because paraphrase relationship is defined in the sense of semantic similarity, we only calculate it with samples from vMF distributions. The loss is defined as,

$$max(0, \delta - dist(x_1, x_2)) + dist(x_1, n_1))+$$
$$max(0, \delta - dist(x_1, x_2)) + dist(x_2, n_2))$$

where $dist$ refers to the distance, x_1 and x_2 are sentences with paraphrase relationship, while x_1 and n_1 are those without paraphrase relationships. The similarity function is the cosine similarity between the mean directions of the semantic variables across K components from the two sentences:

$$dist(x_1, x_2) = cosine(\mu(x_1), \mu(x_2))$$

where $\mu(x_i) = (\mathbf{z}_{sem(i)}^{1\cdots K} \odot \mathbf{z}_{0(i)}^{1\cdots K})$ and $\odot$ is the element-wise product.

Word Position Loss. Following Chen et al. (2019), we keep a word position loss (WPL) to guide the representation learning of the syntactic variable. For both word averaging encoders and LSTM encoders, we parameterize WPL with a three-layer feedforward neural network $f(\cdot)$. The concatenation of the samples of the syntactic variables $\mathbf{z}_{syn}$ and the embedding vector $\mathbf{emb}_i$ at the word position i form the input for the network. In the decoder stage, the position representation at position i is predicted as a one-hot vector. The corresponding equation is defined as,

$$WPL = \mathbb{E}_{z_{syn}\sim q_\phi(z|x)}\Big[\sum_i \log[(f([e_i; z_{syn}]))_i]\Big]$$

where $(\cdot)_i$ is the probability of position i.

Inference Model for Word Averaging. In our framework, syntax and semantics encoders $q_\phi^e(\mathbf{z}_{syn}|\mathbf{x})$ and $q_\phi^e(\mathbf{z}_{sem}|\mathbf{x})$ follow different fashions with different sampling strategies with additional linear feedforward neural network. However, both use word averaging to obtain the mean vector, $\mu(\mathbf{x})$ and the standard deviation vector, $\sigma(\mathbf{x})$.

In the decoding stage, we generate a bag of words given $\mathbf{z}_{syn}$ and $\mathbf{z}_{sem}$ by the posterior probability $p_\theta^d(\mathbf{x}|\mathbf{z}_{syn}, \mathbf{z}_{sem})$. Note that the decoding output is a tuple of vectors, which includes both word index and their component probability distribution. The expected output log-probability is computed as follows:

$$\mathop{\mathbb{E}}_{\substack{\mathbf{z}_{sem}\sim q_\phi^e(\mathbf{z_{sem}}|\mathbf{x})\\ \mathbf{z}_{syn}\sim q_\phi^e(\mathbf{z_{syn}}|\mathbf{x})}}[\log p_\theta^d(\mathbf{x}|\mathbf{z}_{sem}, \mathbf{z}_{syn})] =$$

$$\mathop{\mathbb{E}}_{\substack{\mathbf{z}_{sem}\sim q_\phi^e(\mathbf{z_{sem}}|\mathbf{x})\\ \mathbf{z}_{syn}\sim q_\phi^e(\mathbf{z_{syn}}|\mathbf{x})}}\Big[\sum_{t=1}^{T}\log\frac{\exp f_\theta([\mathbf{z}_{sem};\mathbf{z}_{syn}])_{x_t}}{\sum_{v=1}^{V}\exp f_\theta([\mathbf{z}_{sem};\mathbf{z}_{syn}])_v}\Big]$$

where V is the vocabulary size, [;] indicates concatenation, T is the sentence length and x_t is the index of the t'th word's word type. $f_\theta([\mathbf{z}_{sem}; \mathbf{z}_{syn}]$ is a feedforward neural network with outputs being a bag of words.

Inference Model for BLSTM Averaging

Similarly, we compute the expected output log-probability of generated words, including their component information for BLSTM as follows,

$$\mathop{\mathbb{E}}_{\substack{\mathbf{z}_{sem}\sim q_\phi^e(\mathbf{z_{sem}}|\mathbf{x})\\ \mathbf{z}_{syn}\sim q_\phi^e(\mathbf{z_{syn}}|\mathbf{x})}}[\log p_\theta^d(\mathbf{x}|\mathbf{z}_{sem}, \mathbf{z}_{syn})] =$$

$$\mathop{\mathbb{E}}_{\substack{\mathbf{z}_{sem}\sim q_\phi^e(\mathbf{z_{sem}}|\mathbf{x})\\ \mathbf{z}_{syn}\sim q_\phi^e(\mathbf{z_{syn}}|\mathbf{x})}}\Big[\sum_{w=1}^{S}\log p_\theta(x_w|\mathbf{z}_{syn}, \mathbf{z}_{sem}, \mathbf{x}_{1:s-1}|)\Big]$$

The inference model $q_\phi^e(\mathbf{z}_{sem})$ is still a word averaging encoder while $q_\phi^e(\mathbf{z}_{syn})$ is parameterized by a bidirectional LSTM, where the forward and backward hidden states are concatenated together and then the average is taken. The averages are used as input for a feedforward network with one hidden layer to produce both mean vector $\mu(\mathbf{x})$ and $\sigma(\mathbf{x})$.

Since both the inference model of word averaging and BLSTM are interacting with the decomposed KL divergence or total correlations through backpropagation, our inference and the generative models can obtain more factorized component information. Hence, the generated tokens are more consistent between syntax and semantics.

4 Experiments

Following Chen et al. (2019), we sampled 50M paraphrase pairs from ParaNMT-50M (Wieting and Gimpel, 2018) as our training set. We use the SemEval semantic textual similarity (STS) task 2017 (Cer et al., 2017) as the development set. The STS task and its benchmark as the test set for similarity evaluation. The implementation was based on the PaddlePaddle deep learning platform.

4.1 Experiment Setup

We set the dimension of hidden variables and word embedding to 50, which speeds up experiments and provides a competitive performance over a wide range. To have a fair comparison, we also tune γ, the weights for PRL and reconstruction loss from 0.1 to 1 in increments of 0.1 based on the development set performance. We set $\gamma = 0.2$ with the best validation results. One sample from each latent variable is utilized during training. When evaluating DecVAE based models on STS tasks, the mean direction of the semantic variable is used. In contrast, the mean vector of the syntactic variable is used in syntactic similarity tasks. The total correlations are also mainly applied to syntactic tasks since we find that applying total correlations to vMF distribution makes the model too complicated. Hence, we simplify the framework with only KL divergence of attentions calculated against the semantic components for current work.

4.2 Baselines

We compare with word averaging (WORD$_{\text{AVG}}$) and bidirectional LSTM averaging (BLSTM$_{\text{AVG}}$) of VGVAE model (Chen et al., 2019; Wieting and Gimpel, 2018). In particular, WORD$_{\text{AVG}}$ takes the average over word embeddings in the input sequence to obtain the sentence representation. BLSTM$_{\text{AVG}}$ uses the average hidden states of a bidirectional LSTM as the sentence representation, where forward and backward hidden states are concatenated.

4.3 Semantic Similarity Evaluations

Table 1 presents the semantic similarity evaluations. Specifically, the upper rows tell us how they can model similarity when trained on paraphrases (Wieting and Gimpel, 2018) and the lower half rows show remarkable differences between semantic and syntactic metrics. It is worth noting that in Chen et al. (2019), they also reported semantic modeling results for several pretrained embeddings, in which they showed that all pretrained embeddings are far lower than those of VGVAE based models. Such a result implies that VAE-based modeling can capture semantics quite well no matter what variations we make. For simplicity, we do not show the results from pretrained embeddings herein. Readers please refer to Chen et al. (2019) for more details.

methods	semantic var. %		syntactic var. %	
	bm	avg	bm	avg
VGVAE WORD$_{\text{AVG}}$	71.9	64.8	-	-
VGVAE BLSTM$_{\text{AVG}}$	71.4	64.4	-	-
DecVAE WORD$_{\text{AVG}}$	**72.4**	**65.1**	-	-
DecVAE BLSTTM$_{\text{AVG}}$	71.4	63.2	-	-
VGVAE ALL+LSTM enc	72.2	65.1	16.6	24.3
VGVAE ALL+LSTM e&d	72.8	65.3	11.5	19.9
DecVAE+WPL	52.3	45.3	31.4	33.2
DecVAE+DPL	63.5	57.6	35.9	37.5
DecVAE+PRL	65.6	59.2	28.9	33.1
DecVAE+PRL+WPL	69.9	62.9	24.4	28.2
DecVAE+PRL+DPL	67.5	62.3	34.1	32.8
DecVAE+DPL+WPL	69.9	65.4	19.9	24.2
DecVAE+ALL + WORD$_{\text{AVG}}$ e&d	**73.9**	64.0	22.3	17.7
DecVAE ALL+LSTM enc	70.0	62.1	14.7	16.5
DecVAE ALL+LSTM e&d	72.2	**65.7**	**8.1**	**9.7**

Table 1: Pearson correlation (%) for STS test sets. bm: STS test set. avg: the average of Pearson correlation for each domain in the test set from 2012 to 2016. Results are in bold if they are highest in the "semantic variable" columns or lowest in the "syntactic variable" columns. "ALL" indicates all of the multi-task losses are used. "e&d" means "enc & dec". The results are averaged over five repetitions and the standard deviation is around 0.1%-0.2% for all methods.

As shown in the upper rows of Table 1, DecVAE+WORD$_{\text{AVG}}$ achieves the best semantic score for both STS avg metric and STS bm metric. LSTM-based models do not show advantages over $Word_{AVG}$ as VGVAE (Chen et al., 2019). So average of LSTM outputs for decomposed VAE is not as effective as vanilla VAE based approaches.

The lower rows in Table 1 show whether semantic variables can better capture semantic information than syntactic variables. We reproduced VGVAE's result by their released package (Chen et al., 2019) for comparisons and our results are lines from 3 to 11. As shown, the semantic and syntactic variables of the base DecVAE model show similar performances on the STS test sets. With more losses added, the performance of these two variables gradually diverges, indicating that different information is captured in the two variables. Therefore, we can see that the various losses play essential roles in the disentanglement of semantics and syntax in DecVAE. When all losses plus $Word_{AVGe\&d}$ are fully utilized, the high-

est benchmark results (73.91%) are obtained with 1.7% higher than VGVAE for semantic variables. Meanwhile, all losses plus $LSTM_{e\&d}$ achieves the best average results for semantic variables. More impressively, this approach yields relatively low scores for both benchmarks and average of syntactic variables (8.05 and 9.72 for bm and avg respectively). This fully shows that decomposition with total correlation has excellent disentanglement capacity on semantics and syntax.

Finally, Figure 3 plots the performance curves of our models and baselines as the length of the target sentence increases. We observe a similar trend, i.e., the longer the sentence, the worse the performance. Our framework is close to the top (red) curve and has a more consistent trend. This shows that DecVAE achieves more remarkable disentanglement effects in syntax. Particularly, in Table 1, the full model with LSTM encoder and decoder achieves much lower values for syntactic evaluations than all other models.

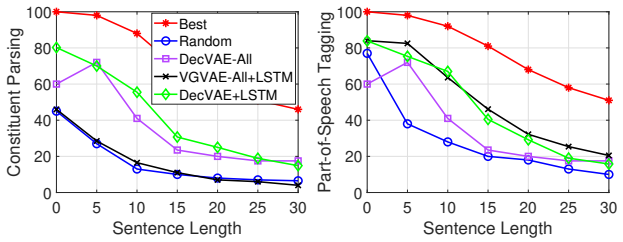

Figure 3: Constituency parsing F1 scores (left) and POS tagging accuracy (right) by sentence length, for 1-nearest neighbor parsers based on semantic and syntactic variables, as well as a random baseline and an oracle nearest neighbor parser ("Best"). Note that in the legend, "+LSTM" means "+LSTM enc & dec".

4.4 Syntactic Similarity Evaluation

Following the evaluation protocol in VGVAE (Chen et al., 2019), we utilize syntactic variables to calculate nearest neighbors for a 1-nearest-neighbor syntactic parser or POS tagger. Several metrics are employed to quantify the quality of the parser's output and tagging sequences. It is worth noting that this evaluation does not directly compare parsing accuracy. Instead, similar to the semantic similarity, it demonstrates syntactic variables' ability to capture more syntactic information than semantic variables.

We report labeled F1 of constituent parsing and accuracy of POS tagging in Table 2. First, we evaluate VGVAE and DecVAE with word averaging encoder and BLSTM encoder in the upper table.

	Constituent Parsing (F_1, ↑).		POS Tagging (% Acc., ↑).	
VGVAE WORDAVG	25.5		21.4	
VGVAE BLSTMAVG	25.7		21.6	
DecVAE WORDAVG	27.8		24.9	
DecVAE BLSTMAVG	29.9		33.2	
	semV.	synV.	semV.	synV.
VGVAE All	25.4	29.3	21.4	25.5
VGVAE+LSTM enc. & dec.	25.3	38.8	21.4	35.7
DecVAE All	24.9	33.7	20.4	29.8
DecVAE+LSTM enc.	24.5	36.9	21.4	35.5
DecVAE+LSTM enc. & dec.	**23.2**	**41.5**	**19.4**	**38.9**

Table 2: Syntactic similarity evaluations, labeled F1 score for constituent parsing, and accuracy (%) for part-of-speech tagging. Numbers are bold if they are worst in the "semantic variable" column or best in the "syntactic variable" column. "ALL" indicates all of the multi-task losses are used. The results are collected and averaged over five rounds and the standard deviation is around 0.1%-0.2% for all methods.

DecVAE outperforms VGVAE in both parsing and tagging. For the lower part, in contrast to semantic similarity, syntactic variables are expected to boost both tasks while semantic variables worsen them. The baseline "VGVAE All" initially have similar results for two variables. Then, with the addition of LSTM encoder and decoder, expected performances appear along. For our method, the gaps between both variables are more remarkable than VGVAE, although not always worst for semantic variables and best for syntactic variables. Such a result indicates that DecVAE achieved a good disentanglement of syntax and semantics. In particular, our full combination with LSTM achieves the best results and outperforms those of SOTA.

Another observation is that although both VGVAE and DecVAE do not perform well compared with their LSTM counterparts, "DecVAE All" still obtains better performances than VGVAE. We believe that it is the total correlation that brings more accurate disentanglement effects. Nonetheless, the syntactic evaluation results, in general, are not so evident as the semantic correspondents.

4.5 Qualitative Analysis with Case Studies

We conduct a qualitative evaluation of latent variables via cosine similarity for nearest neighbor sentences and words to test set examples in terms of both the semantic and syntactic representations. The results are reported in Table 3 and Table 4.

4.5.1 Lexical Analysis

Table 3 shows word nearest neighbors for both semantic and syntactic representations and exhibits

Query Words	Retrieved Words
exact	*semantic*: indeed, current, completely, absolutely, context, clear, strictly, similarly, ec, proper *syntactic*: soap, benefit, license, orn, discontinuation, wed, jin, applications, girls, lucian
command	*semantic*: guidance, result, ec, direction, accept, ordering, release, transmission, order *syntactic*: problem, root, eleven, sex, jinglge, francis, sale, trains, sixteen, industrial
requesting	*semantic*: note, guidance, inquires, inception, accepted, needs, claims, query, required, application *syntactic*: terminate, subscribe, particle, composite, locate, require, claim, compose, apply, inquiring
emptying	*semantic*: changing, reset, stuffed, withdrawn, outline, modified, remove, boo, restoring, threads *syntactic*: entering, obtained, subtotal, living, combine, surged, dismissed, composed, applying, inquiring
smallest	*semantic*: minor, mi, smaller, diffuse, events, types, fragments, size, short, weighing *syntactic*: biggest, odd, stable, concerned, small, hotter, hottest, shorter, fragmentary

Table 3: Examples of most similar words to particular query words in terms of the semantic or syntactic variable

Query Sentence	Semantically Similar	Syntactically Similar
go, you fools, Xar bellowed	the hell, you say, Alekseyv bellowed	Huh, I've got file festivals to enter he said.
Do you think I could do what she did?	Do you think that I'd do it like that?	So, do you know who's there?
His head must be right between the two cuts.	He is already getting in your head right now.	My mom even basked a cake for the party.
I'll tell you things can change a lot.	When the siutation changes, we'll let you know.	I'd like to try the state government again.
They say, you do not have a face.	In fact, you's just a pretty face.	You don't know what is in that building
I even found a rare gouda on the internet.	I've seen a lot on the internet.	Did you get your degree off a cereal box?
I don't know, he was wearing socks.	you got any socks you do not want wear.	you don't play piano, I hope.
I love you as much as before.	I love you more than I ever loved anyone.	but wait. There's as much as what is.
You know what, cal, just pull over.	cal, is trying to pull you out.	You know, you guys got some competition out there?
Yeah, he got punched out in court earlier.	From there she was taken to court and back.	He would have to be forged by Jupityer himself.

Table 4: Examples of most similar sentences to particular query sentences in terms of the semantic or syntactic variable.

clear patterns. Among the five query words, retrieved words based on semantics have similar meanings against them, while those based on syntax share part-of-speeches. For example, for the query word, *exact*, almost all words in the semantic row have the sense of exactness. Likewise, most of the words in the second row, semantically, have the sense of order, as the query word, *command*. In contrast, the syntactic part has POS as *NN*. For the third row, semantically, they mostly have an association with *require* while syntactically, they are all verbs.

4.5.2 Sentential Analysis

Table 4 demonstrates sentences of semantically and syntactic similar respectively in column 2 and column 3. Like the lexical similarity, retrieved sentences in column 2 have similar meanings or similar keywords or key phrases to query sentences while they may be different in sentence structure. For example, "*bellowed*", "*Do you think*", "*head*", "*change*", "*internet*", "*love*", "*pull*" and "*court*" are in the rows from one to ten respectively.

In contrast, those that are syntactically similar may have different meanings while they have similar grammatical patterns. Take a few rows as examples, "*go, you fools, Xar bellowed*" does have similar syntactic construction to "*Huh, I've got file festivals to enter he said*". Likewise, the second row, the query is composed of yes/no questions with an object clause for both query and syntactically similar sentence.

4.6 Discussions

The above results show the disentanglement effects of our proposed DecVAE from semantic and syntactic evaluations in both quantitative and qualitative perspectives. In comparing with baselines, it is not hard to see that DecVAE demonstrates more impressive disentanglement powers. Such results confirm our assumption that a more fine-tuned decomposition of KL divergences can detect more subtle aspects of semantics and syntax. This discovery can shed light on constructing more representative learning strategies for languages in both token and sentence levels.

5 Conclusion

We propose DecVAE, a framework to disentangle syntax and semantics in a sentence. It extends the original VAE so that the latent variables can be separated in more interpretable way. Experiments show that DecVAE achieves better results in semantic and syntax similarity than that of SOTA. One future direction is fine-grained representation learning for words and sentences, which is essential for many downstream applications such as controllable text generation. Besides, continual and interactive feature distillation may help improve more discriminate disentanglement (Wang et al., 2021).

References

David M. Blei and John D. Lafferty. 2006. Dynamic topic models. In *Proceedings of the Twenty-Third International Conference on Machine Learning (ICML)*, pages 113–120, Pittsburgh, PA.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022.

Samuel R. Bowman, Luke Vilnis, Oriol Vinyals, Andrew M. Dai, Rafal Józefowicz, and Samy Bengio. 2016. Generating sentences from a continuous space. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning (CoNLL)*, pages 10–21, Berlin, Germany.

Christopher P. Burgess, Loïc Matthey, Nicholas Watters, Rishabh Kabra, Irina Higgins, Matthew Botvinick, and Alexander Lerchner. 2019. MONet: Unsupervised scene decomposition and representation. *arXiv preprint arXiv:1901.11390*.

Daniel M. Cer, Mona T. Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. Semeval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval@ACL)*, pages 1–14, Vancouver, Canada.

Mingda Chen, Qingming Tang, Karen Livescu, and Kevin Gimpel. 2018. Variational sequential labelers for semi-supervised learning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 215–226, Brussels, Belgium.

Mingda Chen, Qingming Tang, Sam Wiseman, and Kevin Gimpel. 2019. A multi-task approach for disentangling syntax and semantics in sentence representations. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 2453–2464, Minneapolis, MN.

Adji B. Dieng, Chong Wang, Jianfeng Gao, and John W. Paisley. 2017. Topicrnn: A recurrent neural network with long-range semantic dependency. In *Proceedings of the 5th International Conference on Learning Representations (ICLR)*, Toulon, France.

Ran Ding, Ramesh Nallapati, and Bing Xiang. 2018. Coherence-aware neural topic modeling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNL)*, pages 830–836, Brussels, Belgium.

Babak Esmaeili, Hao Wu, Sarthak Jain, Alican Bozkurt, N. Siddharth, Brooks Paige, Dana H. Brooks, Jennifer G. Dy, and Jan-Willem van de Meent. 2019. Structured disentangled representations. In *Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 2525–2534, Naha, Okinawa, Japan.

Zhenxin Fu, Xiaoye Tan, Nanyun Peng, Dongyan Zhao, and Rui Yan. 2018. Style transfer in text: Exploration and evaluation. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence (AAAI)*, pages 663–670, New Orleans, LA.

Irina Higgins, Loïc Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. 2017. beta-vae: Learning basic visual concepts with a constrained variational framework. In *Proceedings of the 5th International Conference on Learning Representations (ICLR)*, Toulon, France.

Shifu Hou, Yujie Fan, Mingxuan Ju, Yanfang Ye, Wenqiang Wan, Kui Wang, Yinming Mei, Qi Xiong, and Fudong Shao. 2021. Disentangled representation learning in heterogeneous information network for large-scale android malware detection in the COVID-19 era and beyond. In *Proceedings of the Thirty-Fifth AAAI Conference on Artificial Intelligence (AAAI)*, pages 7754–7761, Virtual Event.

Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P. Xing. 2017. Toward controlled generation of text. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, pages 1587–1596, Sydney, Australia.

Mohit Iyyer, John Wieting, Kevin Gimpel, and Luke Zettlemoyer. 2018. Adversarial example generation with syntactically controlled paraphrase networks. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 1875–1885, New Orleans, LA.

Sarthak Jain, Edward Banner, Jan-Willem van de Meent, Iain James Marshall, and Byron C. Wallace. 2018. Learning disentangled representations of texts with application to biomedical abstracts. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4683–4693, Brussels, Belgium.

Yeonwoo Jeong and Hyun Oh Song. 2019. Learning discrete and continuous factors of data via alternating disentanglement. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, pages 3091–3099, Long Beach, CA.

Vineet John, Lili Mou, Hareesh Bahuleyan, and Olga Vechtomova. 2019. Disentangled representation learning for non-parallel text style transfer. In *Proceedings of the 57th Conference of the Association for Computational Linguistics (ACL)*, pages 424–434, Florence, Italy.

Mingxuan Ju, Wei Song, Shiyu Sun, Yanfang Ye, Yujie Fan, Shifu Hou, Kenneth A. Loparo, and Liang Zhao. 2021. Dr.emotion: Disentangled representation learning for emotion analysis on social media to improve community resilience in the COVID-19 era and beyond. In *Proceedings of the Web Conference (WWW)*, Virtual Event / Ljubljana, Slovenia.

Hyunjik Kim and Andriy Mnih. 2018. Disentangling by factorising. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, pages 2654–2663, Stockholmsmässan, Stockholm, Sweden.

Diederik P. Kingma and Max Welling. 2014. Auto-encoding variational bayes. In *Proceedings of the 2nd International Conference on Learning Representations (ICLR)*, Banff, Canada.

Dingcheng Li, Siamak Zamani, Jingyuan Zhang, and Ping Li. 2019a. Integration of knowledge graph embedding into topic modeling with hierarchical dirichlet process. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, Minneapolis, MN.

Dingcheng Li, Jingyuan Zhang, and Ping Li. 2018. Representation learning for question classification via topic sparse autoencoder and entity embedding. In *Proceedings of the IEEE International Conference on Big Data (Big Data)*, Seattle, WA.

Dingcheng Li, Jingyuan Zhang, and Ping Li. 2019b. TMSA: A mutual learning model for topic discovery and word embedding. In *Proceedings of the 2019 SIAM International Conference on Data Mining (SDM)*, pages 684–692, Calgary, Alberta, Canada.

Alireza Makhzani and Brendan J. Frey. 2017. Pixelgan autoencoders. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1975–1985, Long Beach, CA.

Yishu Miao, Lei Yu, and Phil Blunsom. 2016. Neural variational inference for text processing. In *Proceedings of the 33nd International Conference on Machine Learning (ICML)*, pages 1727–1736, New York City, NY.

Nikhil Mishra, Mostafa Rohaninejad, Xi Chen, and Pieter Abbeel. 2018. A simple neural attentive meta-learner. In *Proceedings of the 6th International Conference on Learning Representations (ICLR)*, Vancouver, Canada.

Andriy Mnih and Karol Gregor. 2014. Neural variational inference and learning in belief networks. In *Proceedings of the 31th International Conference on Machine Learning (ICML)*, pages 1791–1799, Beijing, China.

Feng Nan, Ran Ding, Ramesh Nallapati, and Bing Xiang. 2019. Topic modeling with wasserstein autoencoders. In *Proceedings of the 57th Conference of the Association for Computational Linguistics (ACL)*, pages 6345–6381, Florence, Italy.

Gabriele Pergola, Lin Gui, and Yulan He. 2021. A disentangled adversarial neural topic model for separating opinions from plots in user reviews. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 2870–2883, Online.

Victor Sanh, Thomas Wolf, and Sebastian Ruder. 2019. A hierarchical multi-task approach for learning embeddings from semantic tasks. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence (AAAI)*, pages 6949–6956, Honolulu, HI.

Akash Srivastava and Charles Sutton. 2017. Autoencoding variational inference for topic models. In *Proceedings the 5th International Conference on Learning Representations (ICLR)*, Toulon, France.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems (NIPS)*, pages 5998–6008, Long Beach, CA.

Yigong Wang, Zhuoyi Wang, Yu Lin, Latifur Khan, and Dingcheng Li. 2021. CIFDM: continual and interactive feature distillation for multi-label stream learning. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pages 2121–2125, Virtual Event, Canada.

Michael Satosi Watanabe. 1960. Information theoretical analysis of multivariate correlation. *IBM J. Res. Dev.*, 4(1):66–82.

John Wieting and Kevin Gimpel. 2018. Paranmt-50m: Pushing the limits of paraphrastic sentence embeddings with millions of machine translations. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 451–462, Melbourne, Australia.

Sam Wiseman, Stuart M. Shieber, and Alexander M. Rush. 2018. Learning neural templates for text generation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3174–3187, Brussels, Belgium.

Zichao Yang, Zhiting Hu, Ruslan Salakhutdinov, and Taylor Berg-Kirkpatrick. 2017. Improved variational autoencoders for text modeling using dilated convolutions. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, pages 3881–3890, Sydney, Australia.

Qingyu Yin, Yu Zhang, Weinan Zhang, Ting Liu, and William Yang Wang. 2018. Deep reinforcement learning for chinese zero pronoun resolution. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 569–578, Melbourne, Australia.

Chunting Zhou and Graham Neubig. 2017. Multi-space variational encoder-decoders for semi-supervised labeled sequence transduction. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 310–320, Vancouver, Canada.