Knowledge-Based Systems 165 (2019) 30–39

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

Semi-supervised dimensional sentiment analysis with variational autoencoder

Chuhan Wu [a,*], Fangzhao Wu [b], Sixing Wu [a], Zhigang Yuan [a], Junxin Liu [a], Yongfeng Huang [a]

[a] Department of Electronic Engineering, Tsinghua University, Beijing 100084, China
[b] Microsoft Research Asia, Beijing 100084, China

ARTICLE INFO

Article history:
Received 11 June 2018
Received in revised form 12 November 2018
Accepted 13 November 2018
Available online 20 November 2018

Keywords:
Dimensional sentiment analysis
Variational autoencoder
Semi-supervised learning

ABSTRACT

Dimensional sentiment analysis (DSA) aims to compute real-valued sentiment scores of texts in multiple dimensions such as valence and arousal. Existing methods for DSA are usually based on supervised learning. However, it is expensive and time-consuming to annotate sufficient samples for training. In this paper, we propose a semi-supervised approach for DSA based on the variational autoencoder model. Our model consists of three modules: an encoding module to encode sentences into hidden vectors, a sentiment prediction module to predict the sentiment scores of sentences, and a decoding module that takes the outputs of the preceding two modules as input and reconstructs the input sentences. In our approach, the sentiment prediction module is encouraged to accurately predict sentiment scores of both labeled and unlabeled texts to help the decoding module reconstruct such texts more accurately. Thus, our approach can exploit useful information in unlabeled data. Experimental results on three benchmark datasets show that our approach can effectively improve the performance of DSA with considerably less labeled data.

1. Introduction

Traditional sentiment analysis methods are primarily designed to classify the sentiment polarity (e.g., positive, negative and neutral) or emotion category (e.g., joy, sad and anger) of texts [1–7]. However, these methods are coarse-grained and may ignore the subtle sentiment distinctions among different texts [8]. To address this problem, dimensional sentiment analysis (DSA) is proposed to mine fine-grained sentiment information from texts [9–11]. In DSA, real-valued sentiment scores of texts are computed in multiple dimensions. Various sentiment dimensions have been proposed. For instance, Russell and Mehrabian [9] proposed a three-dimensional model, known as Valence–Arousal–Dominance (VAD). The three dimensions correspond to polarity (pleasure or unpleasure), intensity (excitement or calm) and perceived degree of control over a situation. Based on VAD, Ressel [10] proposed a simplified model named Valence–Arousal (VA), as shown in Fig. 1 and Table 1.

Existing methods for DSA are primarily based on supervised learning [12,13]. For instance, Malandrakis et al. [12] proposed a non-linear regression method using n-gram features to predict real-valued sentiment scores of sentences. The researchers used semantic similarities to obtain affective ratings of n-gram terms based on a set of annotated seed words. Wang et al. [13] introduced a regional CNN-LSTM model for this task. The authors applied a convolutional neural network (CNN) to capture local information within a region and a long short term memory (LSTM) network to learn long-distance information between different regions. Although these supervised learning-based methods can achieve satisfactory performance on DSA, they require a large number of labeled samples to train models, which usually necessitates expensive and time-consuming manual annotation.

To overcome this challenge, in this paper, we propose a semi-supervised approach to dimensional sentiment analysis based on a variational autoencoder (VAE) [14].[1] In our approach, we propose a variant model of standard semi-supervised VAE [15] for DSA. Our model contains three modules: encoding, sentiment prediction and decoding. The encoding module is used to encode input texts into dense hidden vectors via LSTM. The sentiment prediction module is used to predict sentiment scores of texts in different dimensions via a 2-layer stacked Bi-LSTM. The decoding module takes the outputs of the previous modules as input to generate a latent space and reconstructs the original texts using LSTM and a sample from the latent space. In our approach, to accurately reconstruct both labeled and unlabeled texts by the decoding module, providing high-quality encoding and sentiment scores of such

* Corresponding author.

E-mail addresses: wuch15@mails.tsinghua.edu.cn (C. Wu), wufangzhao@gmail.com (F. Wu), wu-sx15@mails.tsinghua.edu.cn (S. Wu), yuanzg14@mails.tsinghua.edu.cn (Z. Yuan), ljx16@mails.tsinghua.edu.cn (J. Liu), yfhuang@mail.tsinghua.edu.cn (Y. Huang).

[1] Our code is available at: https://github.com/wuch15/SRV-DSA.

https://doi.org/10.1016/j.knosys.2018.11.018
0950-7051/© 2018 Elsevier B.V. All rights reserved.

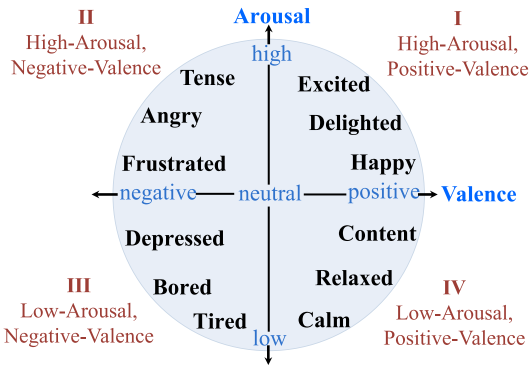


Fig. 1. Representative words in VA model.

Table 1
Several examples with the corresponding VA sentiment scores (between 1 and 9).

Text	V	A
Happy happy happy new year to everybody !!	8.5	8.0
Driving to cape may for the weekend now . So bored ...	3.5	2.0
SICK AGAIN!!!! HATE IT!!!!!	2.5	8.5
Just found out the worst news ever so whatever	2.5	2.0

Table 2
Statistics of datasets.

Dataset	Dimension	Rating	Number of sentence
Facebook	Valence, Arousal	[1,9]	2,896
CVAT	Valence, Arousal	[1,9]	2,009
Emobank	Valence, Arousal, Dominance	[1,5]	10,062

Table 3
Detailed statistics of the CVAT dataset.

Domain	Book	Car	Hotel	Laptop	News	Political
#Sentence	287	256	302	183	542	439

texts is encouraged. Thus, the sentiment prediction module in our approach is required to accurately predict the sentiment scores of both labeled and unlabeled texts. In this way, our approach can exploit the useful information in massive unlabeled data for training the DSA prediction model. Experimental results on three benchmark datasets show that our approach can effectively improve the performance of DSA, especially when training data is insufficient.

2. Related work

2.1. Dimensional sentiment analysis

Dimensional sentiment analysis (DSA) aims to compute real-valued sentiment scores of texts in multiple dimensions [16]. Valence–Arousal–Dominance (VAD) [9] and Valence–Arousal [10] are popular models for describing sentiment dimensions. Existing methods for DSA can be roughly divided into two types: lexicon based methods [17,18] and supervised learning based methods [12, 19].

Lexicon-based methods usually rely on sentiment lexicons to compute the sentiment scores of texts [13]. For example, Kim et al. [17] proposed using the average sentiment ratings of words within a sentence as the sentiment scores of that sentence. Such sentiment ratings of words are obtained from ANEW [20], which is a sentiment lexicon with three-dimensional annotations. Paltoglou et al. [18] improved the aforementioned method by using the weighted arithmetic and geometric mean of the sentiment scores of words. However, the above methods heavily rely on DSA lexicons and cannot capture contextual information of texts.

Supervised learning methods are widely used for DSA [12,19, 21,22]. For instance, Malandrakis et al. [12] proposed a nonlinear regression method to predict sentiment scores of texts. Additionally, the researchers incorporated n-gram features by using a set of seed words to obtain the affective ratings of n-gram terms based on semantic similarity. Buechel and Hahn [19] proposed a linear regression method for DSA. The authors used the average sentiment scores of words within sentences (weighted by term frequency–inverse document frequency) as the input feature of the linear regression model. In recent years, neural network methods, such as multilayer perception [21], boosted neural network [22] and densely connected LSTM [23] were applied to DSA and achieved promising results. For instance, Wang et al. [13] proposed a regional CNN-LSTM architecture for DSA. The researchers applied CNN to extract local information within a region and LSTM to learn long-distance information between regions. The sentiment scores were predicted by a dense layer. However, these supervised learning-based methods require a large number of labeled samples to train their models, which usually necessitates expensive and time-consuming manual annotation. In contrast to these methods, our approach is semi-supervised and can exploit useful information in unlabeled data to reduce the dependence on labeled data.

2.2. Variational autoencoder

Variational autoencoder (VAE) [14] is a generative model that encodes the input into latent variables and generates samples from them [24–27]. VAE was firstly introduced to image generation by Kingma et al. [14]. In the researchers' approach, VAE encodes an input image into a latent space that follows the Gaussian distribution using an encoder and then reconstructs the input image from the latent space using a decoder. Yoo et al. [28] proposed using VAE to predict the real-valued label of images. In the authors' approach, the label of as image is also generated from the latent space. VAE has also been applied to NLP tasks [29–32]. For instance, Bowman et al. [29] firstly introduced VAE to text generation. The researchers applied VAE to encode input texts into a continuous latent space and generate new sentences by decoding the samples in the latent space.

Semi-supervised VAE was firstly introduced to image classification by Kingma et al. [15]. The authors proposed using VAE to reconstruct the input by incorporating predictions from a classifier. Thus, the classifier could utilize the information provided by VAE for both labeled and unlabeled data. Abbasnejad et al. [33] proposed a semi-supervised approach to image classification. The researchers proposed generating samples from multiple latent spaces. Xu et al. [34] proposed a semi-supervised VAE method to text classification using a conditional LSTM architecture. In the authors' approach, the class labels were treated as a "latent variable" so that the model could learn from both labeled and unlabeled data. However, these semi-supervised VAE methods are not suitable for regression tasks, such as DSA, where the label is continuous, rather than categorical. Thus, we propose a variant model of VAE for the DSA task. In contrast to the standard semi-supervised VAE architecture, the latent variables in our model are parameterized from the outputs of both the encoding module and sentiment prediction module. Experimental results on benchmark datasets show that our model can effectively reduce the dependency on labeled data and outperforms several baseline methods.

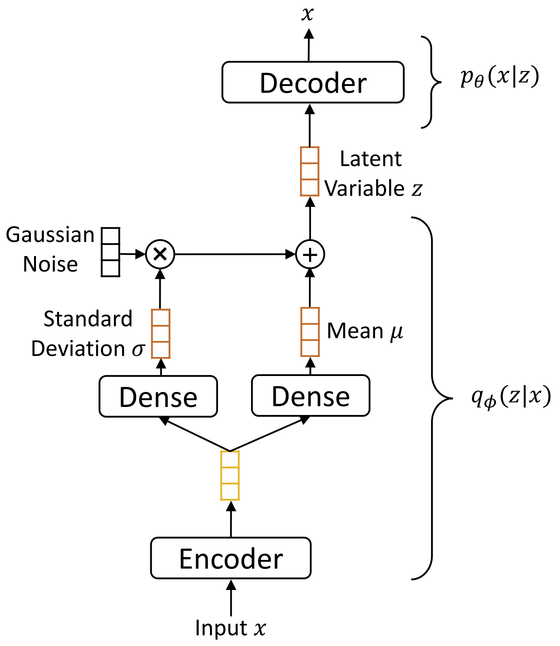


Fig. 2. Architecture of the variational autoencoder.

3. Background

3.1. Variational autoencoder

Variational autoencoder (VAE) is a generative model developed from standard autoencoders [14]. The architecture is shown in Fig. 2. This architecture assumes that a sample x can be generated from a latent variable z. The latent variable z is generated from a prior distribution $p(z)$ (e.g., a standard Gaussian distribution $\mathcal{N}(0, I)$), and sample x can be generated using the conditional distribution $p_\theta(x|z)$, where θ represents the parameters of the decoder. In contrast to standard autoencoders, Kingma et al. [14] proposed approximating the true posterior $p_\theta(z|x)$ by using the encoder to parameterize the mean μ and the standard deviation σ of a diagonal Gaussian matrix $q_\phi(z|x)$ (where ϕ denotes the parameters of the encoder). In this way, samples could be generated by decoding the sampling points in the Gaussian latent space $\mathcal{N}(\mu, diag(\sigma^2))$.

Usually, if the Kullback–Leibler divergence (denoted by KL) between $q_\phi(z|x)$ and $p_\theta(z|x)$ is lower, the approximation of the data distribution is better. Since the true posterior $p_\theta(z|x)$ is unknown, we cannot directly approximate it. To overcome this challenge, the evidence lower bound (ELBO) function is introduced:

$$ELBO = \log p_\theta(x) - KL(q_\phi(z|x)||p_\theta(z|x)), \tag{1}$$

where the first term $\log p_\theta(x)$ is the log-likelihood on x. In this formula, the second term can be minimized by maximizing ELBO because the first term is a constant. ELBO can also be written as follows:

$$ELBO = \mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z)] - KL(q_\phi(z|x)||p(z)), \tag{2}$$

where the first term is the expectation of the conditional log-likelihood of z, and the second term is the Kullback–Leibler divergence between $q_\phi(z|x)$ and $p(z)$, which indicates that VAE expects the learnable posterior to be consistent with the prior $p(z)$. However, sampling from $\mathcal{N}(\mu, diag(\sigma^2))$. directly will lead to a difficulty in end-to-end training. Fortunately, a reparameterization trick has been introduced by Kingma et al. [14] to replace $\mathcal{N}(\mu, diag(\sigma^2))$ by $\mu + \mathcal{N}(0, I) \times \sigma$. In this way, the entire model can be trained in an end-to-end manner.

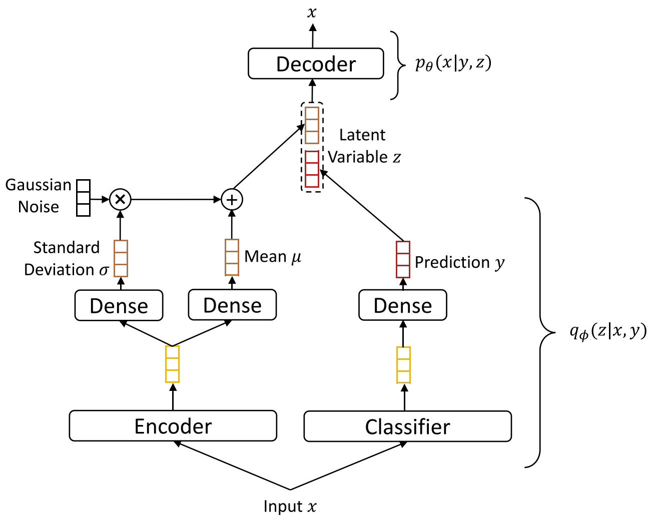


Fig. 3. Architecture of the semi-supervised variational autoencoder.

3.2. Semi-supervised variational autoencoder

Kingma et al. [15] proposed a semi-supervised method based on standard VAE. The architecture is shown in Fig. 3. We refer to this model as SV. In this model, the encoder for generating the latent variable z is formulated as $q_\phi(z|x, y)$, i.e., the latent variable z is parameterized by both x and y. The decoder generates samples from the distribution $p_\theta(x|y, z)$. The label predictive distribution $q_\phi(y|x)$ is given by a classification network. The label y is also regarded as a latent variable and is used to generate a sample x together with z. Thus, the distribution of z is formulated as follows:

$$z \sim q_\phi(z|x, y) = \mathcal{N}(\mu(\tilde{x}, y), diag(\sigma^2(\tilde{x}, y))), \tag{3}$$

where $\tilde{x}$ is the encoded hidden representation of x. Based on the sampled latent variable z, the reconstruction distribution $p_\theta(x|y, z)$ can be formulated as:

$$p_\theta(x|y, z) = D(x|f_d(y, z)), \tag{4}$$

where $f_d(\cdot)$ denotes the function of the decoder, which can be implemented using deep neural networks such as stacked Bi-LSTMs. D represents the distribution of the input data, e.g., a Gaussian or Bernoulli distribution of an image.

Based on this model, Kingma et al. [15] proposed an objective function for semi-supervised VAE based on variational inference. In this model, the evidence lower bound of x with observed label y and corresponding latent variable z can be formulated as:

$$\begin{aligned} \log p_\theta(x, y) \geq \mathbb{E}_{q_\phi(z|x,y)}[\log p_\theta(x|y, z)] + \log p_\theta(y) \\ - KL(q_\phi(z|x, y)||p(z)) = -\mathcal{L}(x, y), \end{aligned} \tag{5}$$

where the term $\mathbb{E}_{q_\phi(z|x,y)}[\log p_\theta(x|y, z)]$ is the expectation of the conditional log-likelihood of latent variable z, and the $\log p_\theta(y)$ term denotes the log-likelihood on y. The term $KL(q_\phi(z|x, y)||p(z))$ is the Kullback–Leibler divergence between the prior distribution $p(z)$ and the posterior distribution $q_\phi(z|x, y)$.

On the unlabeled dataset, the label y is unobserved and is usually given by a classifier. The lower bound of evidence becomes:

$$\begin{aligned} \log p_\theta(x) &\geq \sum_y q_\phi(y|x)(-\mathcal{L}(x, y)) + \mathcal{H}(q_\phi(y|x)) \\ &= -\mathcal{U}(x), \end{aligned} \tag{6}$$

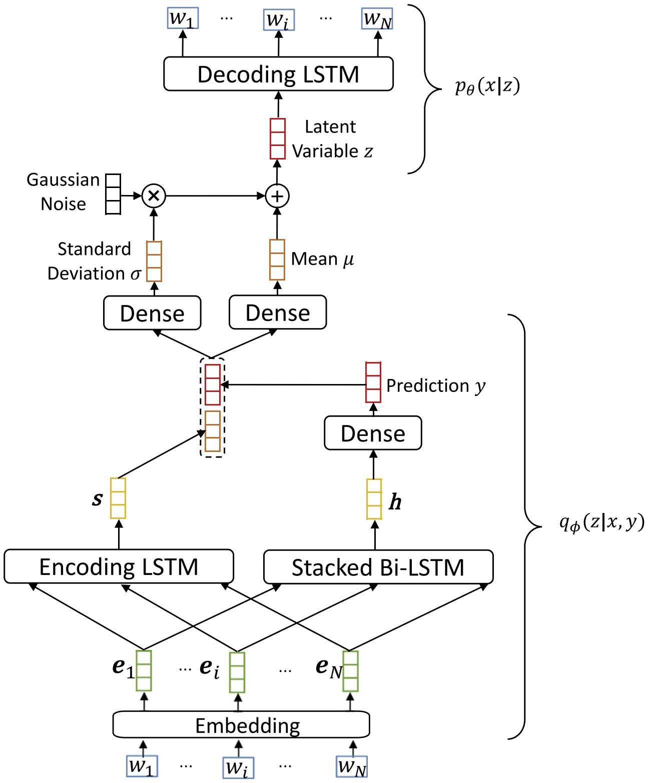


Fig. 4. Architecture of our semi-supervised VAE model.

where $\mathcal{H}$ represents entropy. The final objective function on the entire dataset is:

$$J = \sum_{(x,y)\in S_l} \mathcal{L}(x, y) + \sum_{x\in S_u} \mathcal{U}(x) + \alpha \mathbb{E}_{(x,y)\in S_l}[-\log q_\phi(y|x)] \tag{7}$$

where S_l and S_u represent the labeled and unlabeled datasets respectively. The quantity α is a hyperparameter that controls the relative importance of classification loss on labeled data.

However, since the label y used in Eqs. (6) and (7) is discrete, the semi-supervised VAE method proposed in [15] cannot be directly used in DSA, which is a regression task. Thus, in this paper, we propose a variant model of semi-supervised VAE for DSA, which will be introduced in the next section.

4. Our approach

Based on the semi-supervised VAE model proposed in [15], we propose a variant for the DSA task. Since DSA is a regression task, we name our model semi-supervised regression VAE (SRV). Our model architecture mainly consists of three modules, i.e., encoding, sentiment prediction and decoding, as shown in Fig. 4. Next, we will introduce each module in detail.

4.1. Encoding

The encoding module has two layers. The first layer is an embedding layer, which aims to build the hidden representations of words. It contains two parts, i.e., word embedding and part-of-speech (POS) embedding. The word embedding layer is used to convert a sentence s from a sequence of words $[w_1, w_2, \dots, w_N]$ into a sequence of low-dimensional dense vectors denoted by $[\mathbf{e}_1^w, \mathbf{e}_2^w, \dots, \mathbf{e}_N^w]$. Since sentiment words usually have specific POS tags, such as verb and adjective, we propose using the embedding of the POS tags of corresponding words to enhance word representations. We use the embedding of POS tags rather than the one-hot representation because the latter cannot capture the relations between different POS tags. Similar to word embedding, POS embedding is also a sequence of vectors, denoted by $[\mathbf{e}_1^p, \mathbf{e}_2^p, \dots, \mathbf{e}_N^p]$. We combine word embedding and POS embedding as the output word representations of this layer, i.e., for the ith word, the output is $\mathbf{e}_i = [\mathbf{e}_i^w; \mathbf{e}_i^p]$

The second layer is an encoding LSTM. LSTM has been proven effective in capturing long-distance information [35] and is widely used in the sequence to sequence framework [36]. It takes the embedding sequence $[\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_N]$ of a text as input, and outputs the encoded hidden vectors $\mathbf{s}$ of this text.

4.2. Sentiment prediction

The sentiment prediction module is used to predict the sentiment scores y of texts in multiple dimensions and can parameterize the distribution $q_\phi(y|x)$. It has two parts. The first is a stacked 2-layer Bi-LSTM network. Usually, the overall sentiment of texts depends on both forward and backward contexts. Therefore, scanning the input sequence in both directions may improve sentiment prediction. In addition, since sentiment information can be very complex, a single layer LSTM may be suboptimal. Thus, we apply a stacked 2-layer Bi-LSTM to build the hidden representations $\mathbf{h}$ of the input texts. It shares the same input and the embedding layer with the encoding module.

The second is a dense layer. It will compute the sentiment score $\mathbf{y}$ from $\mathbf{h}$ using a linear function, i.e., $\mathbf{y} = \mathbf{W}\mathbf{h} + \mathbf{b}$, where $\mathbf{W}$ and $\mathbf{b}$ are parameters.

4.3. Decoding

The decoding module is used to reconstruct the input text. The module contains three parts. The first part is a linear transformation module, and it consists of two dense layers that aim to learn the mean vector μ and the standard deviation vector σ. This part takes the outputs of both encoding and sentiment prediction modules, i.e., $[\mathbf{y}; \mathbf{s}]$, as input. Therefore, μ and σ are computed by:

$$\mu = \mathbf{W}_\mu[\mathbf{y}; \mathbf{s}] + \mathbf{b}_\mu, \tag{8}$$

$$\sigma = \mathbf{W}_\sigma[\mathbf{y}; \mathbf{s}] + \mathbf{b}_\sigma, \tag{9}$$

where $\mathbf{W}_\mu$, $\mathbf{W}_\sigma$, $\mathbf{b}_\mu$ and $\mathbf{b}_\sigma$ are parameters.

The second part is a sampling module. It is used to generate the latent variables $\mathbf{z}$ by sampling from the Gaussian distribution $\mathcal{N}(\mu, diag(\sigma^2))$. In this model, since the sentiment scores y are transformed by dense layers first, the latent variable z can be formulated as:

$$z \sim q_\phi(z|x, y) = \mathcal{N}(\mu(\tilde{x}, \tilde{y}), diag(\sigma^2(\tilde{x}, \tilde{y}))), \tag{10}$$

where $\tilde{y}$ denotes the transformed sentiment scores after the dense layers. To train our model in an end-to-end manner, we also use the reparameterization strategy proposed by Kingma et al. [14].

The third part is a decoding LSTM that is used to reconstruct the input text. In this layer, the input of each time step is the output state from the previous step. It is used to parameterize the following distribution:

$$p_\theta(x|z) = D(x|f_d(z)), \tag{11}$$

where $D(\cdot)$ denotes the distribution of the input texts, and $f_d(\cdot)$ denotes the function of the decoding LSTM.

The objective function of our model for the DSA regression task is derived from [15]. Since the sentiment scores y are also used to

Table 4
Performance on the CVAT and Facebook datasets of various methods using various ratios of labeled data. V and A represent valence and arousal, respectively. The percentage denotes the relative improvement of SLSTM.

Model	CVAT						Facebook					
	10%		20%		40%		10%		20%		40%	
	V	A	V	A	V	A	V	A	V	A	V	A
CNN	.519	.287	.585	.318	.599	.344	.389	.648	.521	.696	.564	.728
LSTM	.180	.027	.338	.163	.572	.274	.414	.747	.543	.803	.582	.845
Bi-LSTM	.447	.144	.525	.256	.613	.339	.429	.754	.566	.810	.599	.853
SLSTM	.547	.296	.618	.338	.676	.387	.466	.823	.589	.861	.620	.883
Wang et al. [13]	.532	.285	.614	.330	.670	.381	.453	.806	.575	.845	.601	.866
ST-SLSTM	.555(1.5%)	.299(1.0%)	.626(1.3%)	.348(3.0%)	.683(1.0%)	.394(1.8%)	.480(3.0%)	.836(1.6%)	.594(0.8%)	.866(0.6%)	.624(0.6%)	.885(0.2%)
CT-SLSTM	.564(3.1%)	.313(5.7%)	.632(2.3%)	.357(5.6%)	.690(2.1%)	.399(3.1%)	.505(8.4%)	.847(2.9%)	.611(3.7%)	.873(1.4%)	.640(3.2%)	.888(0.6%)
SV-SLSTM	.578(5.7%)	.330(11.5%)	.651(5.3%)	.364(7.7%)	.702(3.8%)	.405(4.7%)	.518(11.2%)	.854(3.8%)	.632(7.3%)	.878(2.0%)	.664(7.1%)	.897(1.6%)
SRV-SLSTM	.596(9.0%)	.358(20.9%)	.671(8.6%)	.380(12.4%)	.713(5.5%)	.419(8.3%)	.540(15.9%)	.876(6.4%)	.652(10.7%)	.889(3.3%)	.678(9.4%)	.909(2.9%)

generate z, we reformulate the lower bound of evidence as follows:

$$-\mathcal{L}(x,y) = -KL(q_\phi(z|x,y)||p(z|y)) + \log p_\theta(y) + \mathbb{E}_{q_\phi(z|x,y)} \log p_\theta(x|z). \tag{12}$$

However, since DSA is a regression task, the meaning of the term $q_\phi(y|x)$ is different from the classification task in [15], and it is difficult to derive an evidence lower bound of $\log p_\theta(x)$. Hence, for unlabeled data, the loss function is $\mathcal{L}(x,y)$, in which the sentiment score y is given by the sentiment prediction module. For labeled data, there is an additional term for regression loss. Thus, the final objective function of our model is:

$$J = \sum \mathcal{L}(x,y) + \alpha \sum_{(x,y)\in S_l} \mathcal{F}(y), \tag{13}$$

where α is a hyperparameter used to control the relative importance of regression loss on labeled data, and $\mathcal{F}(y)$ is the regression loss function. In our approach, we use the mean absolute error (MAE) as the regression loss.

In the semi-supervised VAE architecture (SV) proposed in [15], the label y is also regarded as a "latent variable" and is combined with z to reconstruct the input sample x. In contrast to the method of [15], in our model SRV, the latent space z in the decoding module is learned from both input x and sentiment scores y. Therefore, the latent variables can encode both sentiment and semantic information of texts in the DSA task, which may be beneficial for building a high quality latent space. In addition, according to the findings of other researchers [8], the sentiment scores are usually unbalanced and do not follow the Gaussian distribution, which will lead to additional KL divergence in model SV. Therefore, regarding y as latent variables may be a suboptimal approach to approximating the distribution of input. Thus, the generated latent space may be smoother if we learn it from both sentiment ratings and text encodings in our SRV model. In our model, the sentiment score y of unlabeled data is predicted by the sentiment prediction module, and the decoding module relies on such predicted sentiment scores, as well as text encodings, to reconstruct the input text. Therefore, to better reconstruct the input text, the sentiment prediction module is encouraged to make more accurate sentiment predictions on both labeled and unlabeled texts. Thus, our approach can exploit useful information in unlabeled data to train models for DSA.

5. Experiments

5.1. *Experimental settings and datasets*

We conducted experiments on three benchmark datasets: the Facebook post dataset provided by Preotiuc-Pietro et al. [37] (referred to as *Facebook*), the Chinese VA dataset constructed by Yu et al. [8] (referred to as *CVAT*), and the Emobank dataset provided by Buechel and Hahn [38]. The details of these datasets are shown in Table 2. Additionally, texts in the CVAT dataset belong to six domains; the detailed statistics are shown in Table 3. In these datasets, the distributions of sentiment scores are usually non-Gaussian. For instance, the marginal distributions of valence and arousal in the Facebook dataset are shown in Fig. 5. Since the marginal distributions of Gaussian variables are also Gaussian, the joint distribution of **y** cannot be Gaussian.

In our experiments, we use 10% of labeled data for testing, and 40% of labeled data for training. The remaining 50% of labeled data is regarded as unlabeled data for semi-supervised learning. We follow the metric used by Mohammad et al. [39] for the real-valued sentiment analysis task. The performance is evaluated by the Pearson correlation coefficient (PCC), computed as follows:

$$PCC = \frac{1}{n-1}\sum_{i=1}^{n}(\frac{A_i - \bar{A}}{\sigma_A})(\frac{P_i - \bar{P}}{\sigma_P}), \tag{14}$$

where A_i is the actual value, P_i is the predicted value, n is the number of test samples, $\bar{A}$ and $\bar{P}$ denote the arithmetic mean of A and P respectively, σ_A and σ_P denote the standard deviation. Usually, a higher PCC value indicates better prediction performance.

We use the Stanford Parser tool[2] to tokenize and obtain POS tags of English sentences. For Chinese sentences, we use the ANSJ segment tool[3] for segmentation and POS tagging. Additionally, since we do not have a rich traditional Chinese corpus, we translate the traditional Chinese sentences into simplified Chinese first.

For the English DSA task, we use the pre-trained embedding weights released by Mikolov et al. [40]. For the Chinese DSA task, we use the open source word2vec tool[4] to train word embedding using the SogouCA News corpus.[5] We train the Chinese word embedding with the skip-gram model. We set the window to 5 and the learning rate to 0.025. All embedding weights are fine-tuned during network training.

In our experiment, the word embedding dimension is set to 300, and the POS embedding dimension is set to 50. The hidden states of encoding and decoding LSTM are 100-dimensional. The hidden states of Bi-LSTM layer are 2 × 100-dim. To prevent overfitting, we apply dropout at rate 0.5 after embedding and all layers. The dimension of the latent variable z is set to 100. The hyper-parameter α in Eq. (13) is set to $0.5\frac{|S|}{|S_l|}$ for all datasets, in which $|S|$ is the size of entire training set and $|S_l|$ is the size of labeled dataset. Our experiments are implemented with the Keras framework [41]. The training is stopped if no improvement on the validation set has been achieved over 5 epochs. We repeated each experiment 10 times and reported the average results.

5.2. *Performance evaluation*

In this section, we compare the performance of our model and several baseline methods trained by 10%, 20% and 40% of labeled data on the three datasets. The fraction of unlabeled data we use for semi-supervised learning is 50%. The methods being compared include:

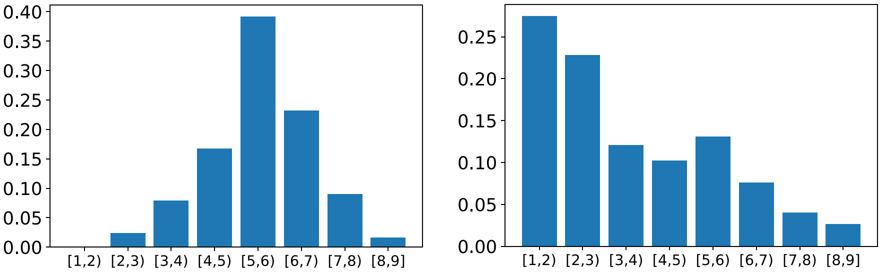
- CNN, using CNN with a global average pooling layer and a dense layer in the prediction module. The CNN has 128 kernels with size 3 and we use global average pooling to obtain the squeezed vectors.
- LSTM, using a single LSTM with a dense layer in the prediction module.
- Bi-LSTM, using a bi-directional LSTM with a dense layer.
- SLSTM, using a 2-layer stacked bi-directional LSTM with a dense layer.
- The method of Wang et al. [13], using the regional CNN-LSTM proposed by Wang et al. However, the researchers' original method can only be applied to document-level DSA (a sentence is a region). For sentence-level DSA, we regard a word as a region.
- ST-SLSTM, using model SLSTM in the prediction module and self training (referred to as ST) as the semi-supervised learning method (iterating until no improvement on the validation set).
- CT-SLSTM, using model SLSTM in the prediction module and the modified co-training (referred to as CT) method proposed by Zhou and Ming [42] for regression.
- SV-SLSTM, using model SLSTM in the prediction module of SV. Eqs. (5) and (13) are used as the objective function to adapt to the regression task.

[2] https://nlp.stanford.edu/software/lex-parser.shtml.

[3] https://github.com/NLPchina/ansj_seg.git.

[4] https://code.google.com/archive/p/word2vec/.

[5] https://www.sogou.com/labs/resour.



(a) Marginal distributions of valence.

(b) Marginal distributions of arousal.

Fig. 5. Marginal distributions of valence and arousal on the Facebook dataset.

Table 5
Performance on the Emobank dataset of various methods using various ratios of labeled data. D represents the dominance dimension. The percentage denotes the relative improvement of SLSTM.

Model	10%			20%			40%		
	V	A	D	V	A	D	V	A	D
CNN	.390	.317	.158	.428	.342	.173	.471	.372	.199
LSTM	.400	.344	.199	.518	.398	.250	.593	.472	.301
Bi-LSTM	.416	.340	.196	.527	.404	.254	.596	.478	.306
SLSTM	.485	.358	.212	.545	.434	.268	.605	.490	.307
Wang et al. [13]	.472	.346	.204	.526	.411	.253	.600	.484	.299
ST-SLSTM	.491(1.2%)	.363(1.4%)	.218(2.8%)	.550(0.9%)	.441(1.6%)	.272(1.5%)	.608(0.5%)	.494(0.8%)	.308(0.3%)
CT-SLSTM	.498(2.7%)	.372(3.9%)	.222(4.7%)	.554(1.7%)	.445(2.5%)	.274(2.2%)	.610(0.8%)	.493(0.6%)	.310(1.70%)
SV-SLSTM	.513(5.8%)	.380(6.1%)	.237(11.8%)	.566(3.9%)	.452(4.1%)	.278(3.7%)	.613(1.3%)	.496(1.2%)	.320(4.2%)
SRV-SLSTM	.530(9.3%)	.397(10.9%)	.254(19.8%)	.578(6.1%)	.460(6.0%)	.294(9.7%)	.620(2.5%)	.508(3.7%)	.333(8.5%)

- SRV-SLSTM, using model SLSTM in the prediction module of SRV.

These methods use the same word and POS embedding as inputs, but the supervised methods (CNN, LSTM, Bi-LSTM and SLSTM) are trained on labeled data only.

The performance of these methods are shown in Tables 4 and 5. The results lead to several major observations.

First, comparing the results using different ratios of labeled data, the performance is better when more labeled samples are used for training. Since the prediction module can learn from labeled data directly, the performance of the entire model can be improved by incorporating more supervised information.

Second, semi-supervised methods consistently outperform the supervised baselines such as SLSTM and the approach of Wang et al. [13]. In addition, the advantage of semi-supervised methods becomes more for smaller quantities of labeled data. Usually, when labeled data is scarce, the representations of texts may be poor. Thus, it is difficult to construct a robust DSA model based on limited labeled data only. Since semi-supervised methods can utilize additional information from unlabeled data, the text representations can be enhanced, which helps predict the sentiment scores more accurately.

Third, the VAE-based methods (i.e., SV and SRV) usually outperform the self-training and co-training methods. Since the text reconstruction relies on the output of the prediction module, the decoding module will encourage the prediction module to provide better predictions. Thus, the semi-supervised models can mine more valuable information from raw text, which can improve the model performance and reduce dependency on labeled data.

Fourth, compared to model SV proposed by Kingma et al. [15], the experimental results show that our model SRV outperforms the original model SV in this task. This may occur because in our architecture, the decoding module generates the latent variable z based on x and y, instead of using x directly. The latent space will contain richer sentiment information and can help the decoding module construct the input texts more accurately. Thus, the performance of prediction module can also be improved. In addition, the distribution of sentiment score y in this task may be unbalanced and does not follow Gaussian distribution, which will lead to additional KL divergence. Thus, regarding y as “latent variables” may be suboptimal for approximating the data distribution. Our model can learn a smooth representation of both semantic and sentiment information by encoding them jointly into latent space.

5.3. *Influence of model architecture*

In this section, we explore the influence of the model architecture, i.e., use different neural networks in the sentiment prediction module. We compare the performance of our SRV approach using CNN, LSTM, Bi-LSTM and SLSTM with various ratios of labeled data. The results on the CVAT dataset are illustrated in Fig. 6. According to the results, using Bi-LSTM in the sentiment prediction module can outperform CNN and single-directional LSTM. This is probably because using Bi-LSTM can capture the contextual information in both directions, which may be useful for accurate sentiment score prediction. In addition, using a stacked Bi-LSTM network in our approach can perform better than using CNN, LSTM and Bi-LSTM. This may occur because using a stacked Bi-LSTM can capture contexts more effectively by building better contextual representations of words. Thus, the sentiment scores can be predicted more accurately.

5.4. *Influence of labeled data*

In this section, we explore the influence of the amount of labeled data. We compare the performance of different methods using different amount of labeled data. For semi-supervised methods, we regard all unused samples in the training set as unlabeled data. The results are illustrated in Fig. 7. According to the results, the advantage of our approaches increases when training data is scarce. These results indicate that standard supervised methods are highly

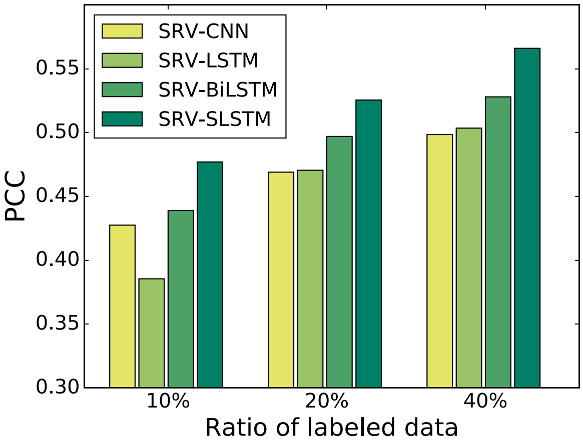


Fig. 6. The influence of model architectures on our approach for the CVAT dataset.

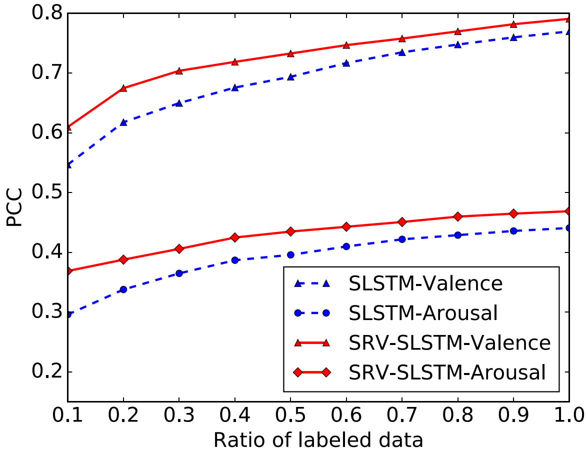


Fig. 7. PCC curves of valence and arousal using various ratios of labeled data for the CVAT dataset.

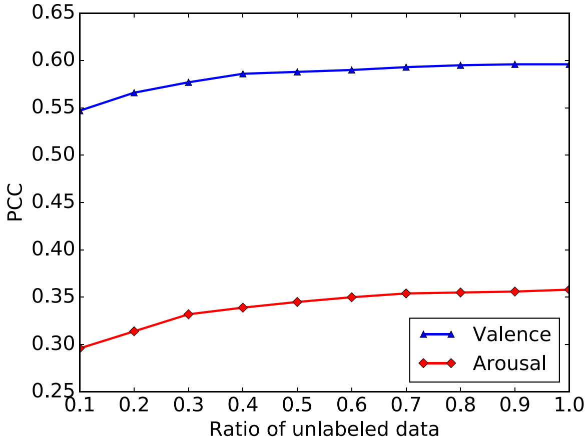


Fig. 8. PCC curves of valence and arousal at various ratios of unlabeled data for the CVAT dataset.

dependent on labeled data. By exploiting the useful information from unlabeled data, our approaches can improve the performance of DSA models and reduce the dependence on labeled data. In addition, when all labeled data is incorporated, our SRV-SLSTM approach can still outperform SLSTM. This probably occurs because of the utilization of useful information in text generation in our approach, since the sentiment prediction module is encouraged to provide accurate predictions to help the decoder reconstruct the input texts.

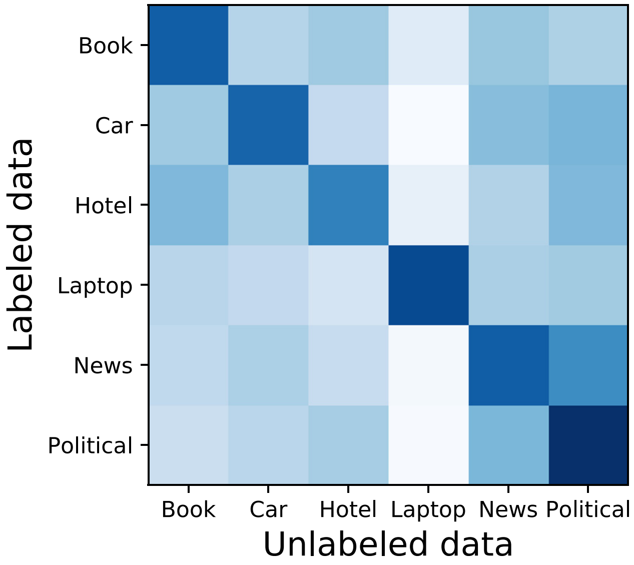


Fig. 9. The improvements brought by using unlabeled data from different domains.

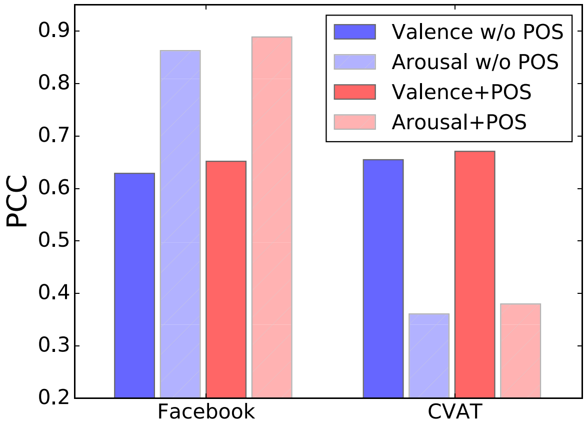
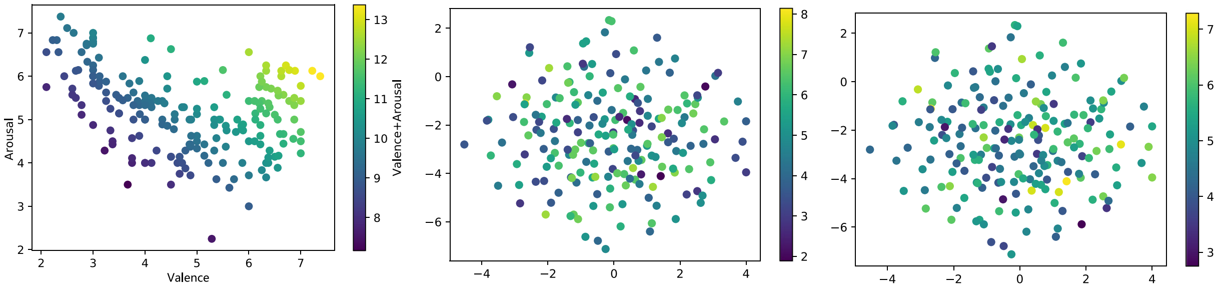


Fig. 10. Effectiveness of POS tags as additional features.

5.5. *Influence of unlabeled data*

We compare the PCC of valence and arousal using 10% of labeled data and various ratios of unlabeled data for semi-supervised learning in our SRV-SLSTM approach. Fig. 8 shows the PCC of our semi-supervised model on the CVAT dataset. The blue and red curves represent valence and arousal, respectively. The performance of both valence and arousal can be effectively improved by semi-supervised learning. This probably occurs because our model can exploit unsupervised information from unlabeled data since the labeled data is highly limited. In addition, the performance will be better if more unlabeled data is used in our experiments. This finding indicates that the unsupervised information learned by text construction is very useful for predicting sentiment scores. The decoding module will perform better in text construction if more unlabeled samples are used. Then it will also require the prediction module to predict the sentiment scores more accurately. Therefore in this way, our model can effectively reduce the dependency on labeled data.



(a) Sentiment scores of test data in VA plane.

(b) Valence of latent variable.

(c) Arousal of latent variable.

Fig. 11. Visualization of samples in the latent space using t-SNE and their sentiment scores in the valence–arousal (VA) plane.

5.6. Influence of the distribution of unlabeled data

To investigate the influence of the distribution of unlabeled data on our approach, we perform several experiments on the CVAT dataset. To obtain different distributions of unlabeled data, we use unlabeled data from different domains (assuming that data in different domains has different distributions). We use 50% of labeled data in each domain and evaluate the performance of our approach when using the same number of unlabeled data from various domains. Fig. 9 shows the relative improvements of PCC (using the average PCC of valence and arousal) of our semi-supervised model using unlabeled data from various domains. Darker colors denote more improvements. According to the results, our semi-supervised learning approach can effectively improve the performance in the DSA task when the unlabeled data originates from most domains in our experiments. This improvement may occur because our semi-supervised model can learn additional information from unlabeled data to help predict the sentiment scores better when labeled data is scarce. In addition, our model can improve more if unlabeled data originates from the same domain as that of labeled data. This improvement may occur because the unlabeled data has the same or similar distribution as that of labeled data, and labeled data, and the posterior $p_\theta(z|x, y)$ can be approximated more accurately.

5.7. Effectiveness of POS feature

We compare the influence of POS features on our approach for the CVAT and Facebook datasets. The semi-supervised training is also conducted on 20% of labeled data of the dataset. The effect of POS embedding feature is shown in Fig. 10. We observe an improvement by adding POS embedding. The experimental results show that POS tags can enrich the linguistic information. For instance, sentiment words are important for identifying the sentiment scores of texts. Usually, these words are adjectives, adverbs or verbs. POS tags can help the model identify such information and predict sentiment scores more accurately.

5.8. Semi-supervised cross-domain DSA

A possible application of our semi-supervised model is cross-domain DSA. Since it is too difficult to annotate sufficient labeled samples for every domain, labeled data may not be sufficient in specific target domains. To overcome this obstacle, we can apply our semi-supervised model to the cross-domain DSA task. Sentences in the CVAT dataset belong to six different domains; hence, we use the labeled data from five source domains and test on the remaining target domain. In comparison, our experiments show results of our model using direct domain transformation (training by labeled data from only source domains) and semi-supervised domain transformation (using labeled data from source domains and 50% of unlabeled data from the target domain). The results are shown in Table 6. The first row in each domain block shows results of the direct domain transformation (denoted by DirectTrans), and the second row corresponds to semi-supervised domain transformation (denoted by SVR). The performance of direct domain transfer in most domains is lower than the previous results, even though more labeled data is used, and some results such as that in the laptop domain are very poor. This finding indicates that certain domains may have very specific features and it is difficult for supervised methods to transfer information without using additional labeled data. However, when using the semi-supervised method by learning from unlabeled data in the target domains, performance can be effectively improved and can reach satisfactory results in most domains. This improvement probably occurs because our semi-supervised model relies less on the labeled data and can exploit useful domain-specific information from unlabeled data in the target domains.

Table 6
Comparison of cross-domain results between direct transform and semi-supervised transform.

Target domain	Method	V	A
Book	DirectTrans	.585	.255
	SVR	.630	.333
Car	DirectTrans	.432	.289
	SVR	.478	.312
Hotel	DirectTrans	.726	.340
	SVR	.745	.360
Laptop	DirectTrans	.544	.145
	SVR	.568	.301
News	DirectTrans	.637	.296
	SVR	.667	.314
Political	DirectTrans	.660	.499
	SVR	.682	.534

5.9. Visualization of latent space

We use 10% of data of the CVAT dataset to visualize the latent space. We observe that the sentiment scores in the VA plane is very unbalanced as shown in Fig. 11(a). The distribution of y is non-Gaussian, which is different from that of z. Therefore in model SV, regarding y as a "latent variable" may make the latent space not smooth. We use t-SNE to visualize the samples in the latent space by representing their VA ratings as colors. Fig. 11(b) and 11(c) show the distribution of valence and arousal, respectively, in the latent space of model SRV. The distribution of samples appears to be Gaussian and the sentiment score is approximately random in the latent space, which means that the dense layers after encoder learn a smooth representation of both sentiment and semantic features.

6. Conclusion

In this paper, we propose a semisupervised approach based on a variational autoencoder for dimensional sentiment analysis. Our approach introduces a variant of VAE with its variational inference to DSA. Our model consists of three modules: an encoding module to encode texts into hidden vectors, a sentiment prediction module to predict the dimensional sentiment scores of texts in multiple dimensions, and a decoding module that takes both outputs of the two preceding modules to reconstruct the input sentences. The experimental results show that our approach can effectively reduce the dependence on labeled data and can effectively improve the performance of the DSA task.

Acknowledgments

This work was supported in part by the National Key Research and Development Program of China under Grant 2016YFB0800402 and in part by the National Natural Science Foundation of China under Grant U1705261, Grant U1536207, Grant U1536201 and U1636113.

References

[1] B. Pang, L. Lee, S. Vaithyanathan, Thumbs up?: Sentiment classification using machine learning techniques, in: Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing, vol. 10, Association for Computational Linguistics, 2002, pp. 79–86.

[2] B. Liu, Sentiment analysis and opinion mining, Synth. Lect. Hum. Lang. Technol. 5 (1) (2012) 1–167.

[3] V. Loia, S. Senatore, A fuzzy-oriented sentic analysis to capture the human emotion in web-based content, Knowl.-Based Syst. 58 (2014) 75–85.

[4] Y. Wang, Y. Rao, X. Zhan, H. Chen, M. Luo, J. Yin, Sentiment and emotion classification over noisy labels, Knowl.-Based Syst. 111 (2016) 207–216.

[5] L. Gui, Y. Zhou, R. Xu, Y. He, Q. Lu, Learning representations from heterogeneous network for sentiment classification of product reviews, Knowl.-Based Syst. 124 (2017) 34–45.

[6] G. Lee, J. Jeong, S. Seo, C. Kim, P. Kang, Sentiment classification with word localization based on weakly supervised learning with a convolutional neural network, Knowl.-Based Syst..

[7] Z. Yuan, S. Wu, F. Wu, J. Liu, Y. Huang, Domain attention model for multi-domain sentiment classification, Knowl.-Based Syst..

[8] L.-C. Yu, L.-H. Lee, S. Hao, J. Wang, Y. He, J. Hu, K.R. Lai, X.-j. Zhang, Building chinese affective resources in valence-arousal dimensions, in: HLT-NAACL, 2016, pp. 540–545.

[9] J.A. Russell, A. Mehrabian, Evidence for a three-factor theory of emotions, J. Res. Personal. 11 (3) (1977) 273–294.

[10] J. Ressel, A circumplex model of affect, J. Personal. Soc. Psychol. 39 (1980) 1161–1178.

[11] L.-C. Yu, L.-H. Lee, K.-F. Wong, Overview of the ialp 2016 shared task on dimensional sentiment analysis for chinese words, in: Asian Language Processing (IALP), 2016 International Conference on, IEEE, 2016, pp. 156–160.

[12] N. Malandrakis, A. Potamianos, E. Iosif, S. Narayanan, Distributional semantic models for affective text analysis, IEEE Trans. Audio, Speech, Lang. Process. 21 (11) (2013) 2379–2392.

[13] J. Wang, L.-C. Yu, K.R. Lai, X. Zhang, Dimensional sentiment analysis using a regional cnn-lstm model, in: ACL 2016—Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, vol. 2, Berlin, Germany, 2016, pp. 225–230.

[14] D.P. Kingma, M. Welling, Auto-encoding variational bayes, arXiv preprint arXiv:1312.6114.

[15] D.P. Kingma, S. Mohamed, D.J. Rezende, M. Welling, Semi-supervised learning with deep generative models, in: Advances in Neural Information Processing Systems, 2014, pp. 3581–3589.

[16] L.-C. Yu, J. Wang, K.R. Lai, X.-j. Zhang, Predicting valence-arousal ratings of words using a weighted graph method, in: ACL, vol. 2, 2015, pp. 788–793.

[17] S.M. Kim, A. Valitutti, R.A. Calvo, Evaluation of unsupervised emotion models to textual affect recognition, in: Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text, Association for Computational Linguistics, 2010, pp. 62–70.

[18] G. Paltoglou, M. Theunis, A. Kappas, M. Thelwall, Predicting emotional responses to long informal text, IEEE Trans. Affect. Comput. 4 (1) (2013) 106–115.

[19] S. Buechel, U. Hahn, Emotion analysis as a regression problem-dimensional models and their implications on emotion representation and metrical evaluation, in: ECAI, 2016, pp. 1114–1122.

[20] M.M. Bradley, P.J. Lang, Affective norms for english words (anew): Instruction manual and affective ratings, Tech. Rep., Citeseer, 1999.

[21] W.-C. Chou, C.-K. Lin, Y.-R. Wang, Y.-F. Liao, Evaluation of weighted graph and neural network models on predicting the valence-arousal ratings of chinese words, in: Asian Language Processing (IALP), 2016 International Conference on, IEEE, 2016, pp. 168–171.

[22] S. Du, X. Zhang, Aicyber's system for ialp 2016 shared task: Character-enhanced word vectors and boosted neural networks, in: Asian Language Processing (IALP), 2016 International Conference on, IEEE, 2016, pp. 161–163.

[23] C. Wu, F. Wu, Y. Huang, S. Wu, Z. Yuan, Thu_ngn at ijcnlp-2017 task 2: Dimensional sentiment analysis for chinese phrases with deep lstm, in: Proceedings of the IJCNLP 2017, in: Shared Tasks, 2017, pp. 47–52.

[24] E. Mansimov, E. Parisotto, J.L. Ba, R. Salakhutdinov, Generating images from captions with attention, arXiv preprint arXiv:1511.02793.

[25] C.K. Sønderby, T. Raiko, L. Maaløe, S.K. Sønderby, O. Winther, Ladder variational autoencoders, in: Advances in Neural Information Processing Systems, 2016, pp. 3738–3746.

[26] Y. Pu, Z. Gan, R. Henao, X. Yuan, C. Li, A. Stevens, L. Carin, Variational autoencoder for deep learning of images, labels and captions, in: Advances in Neural Information Processing Systems, 2016, pp. 2352–2360.

[27] L. Mescheder, S. Nowozin, A. Geiger, Adversarial variational bayes: Unifying variational autoencoders and generative adversarial networks, arXiv preprint arXiv:1701.04722.

[28] Y. Yoo, S. Yun, H.J. Chang, Y. Demiris, J.Y. Choi, Variational autoencoded regression: High dimensional regression of visual data on complex manifold, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 3674–3683.

[29] S.R. Bowman, L. Vilnis, O. Vinyals, A.M. Dai, R. Jozefowicz, S. Bengio, Generating sentences from a continuous space, arXiv preprint arXiv:1511.06349.

[30] Y. Miao, L. Yu, P. Blunsom, Neural variational inference for text processing, in: International Conference on Machine Learning, 2016, pp. 1727–1736.

[31] Z. Hu, Z. Yang, X. Liang, R. Salakhutdinov, E.P. Xing, Controllable text generation, arXiv preprint arXiv:1703.00955.

[32] S. Semeniuta, A. Severyn, E. Barth, A hybrid convolutional variational autoencoder for text generation, arXiv preprint arXiv:1702.02390.

[33] M.E. Abbasnejad, A. Dick, A. van den Hengel, Infinite variational autoencoder for semi-supervised learning, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR, IEEE, 2017, pp. 781–790.

[34] W. Xu, H. Sun, C. Deng, Y. Tan, Variational autoencoder for semi-supervised text classification, in: AAAI, 2017, pp. 3358–3364.

[35] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (7553) (2015) 436.

[36] D. Bahdanau, K. Cho, Y. Bengio, Neural machine translation by jointly learning to align and translate, arXiv preprint arXiv:1409.0473.

[37] D. Preotiuc-Pietro, H.A. Schwartz, G.J. Park, J.C. Eichstaedt, M.L. Kern, L.H. Ungar, E. Shulman, Modelling valence and arousal in facebook posts, in: WASSA@ NAACL-HLT, 2016, pp. 9–15.

[38] S. Buechel, U. Hahn, Emobank: Studying the impact of annotation perspective and representation format on dimensional emotion analysis, in: EACL, 2017, p. 578.

[39] S.M. Mohammad, F. Bravo-Marquez, Wassa-2017 shared task on emotion intensity, arXiv preprint arXiv:1708.03700.

[40] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient estimation of word representations in vector space, arXiv preprint arXiv:1301.3781.

[41] F. Chollet, et al., Keras, 2015.

[42] Z.-H. Zhou, M. Li, Semi-supervised regression with co-training, in: IJCAI, vol. 5, 2005, pp. 908–913.