JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 39, 305-321 (2023)
DOI: 10.6688/JISE.202303_39(2).0004

# Scarce Resource Dimensional Sentiment Analysis Using Domain-Distilled BERT*

WEI LIN AND LIANG-CHIH YU+
*Department of Information Management*
*Yuan Ze University*
*Taoyuan, 320 Taiwan*
*E-mail: lcyu@saturn.yzu.edu.tw+*

Considerable research has focused on dimensional sentiment analysis, which seeks to predict a real-valued sentiment score in multiple dimensions for a given sentiment expression. Although state-of-the-art methods can obtain decent results with high-quality and large-scale corpora data, performance declines significantly under conditions of data scarcity. To address this data scarcity problem, this study proposes a domain-distilled method to learn domain-invariant features instead of the domain-specific features commonly used by traditional methods because learning domain-specific features under data scarcity condition may restrict coverage of the domain feature space. The proposed distillation process is accomplished using a domain discriminator to distinguish the feature's domain. In addition, the domain discriminator is trained by maximizing the prediction loss because this makes it difficult for the discriminator to distinguish among domains, thus improving its ability to learn domain-invariant features. To evaluate the proposed method, we implement the domain-distilled method in Bidirectional Encoder Representations from Transformers (BERT) due to its promising results in many natural language processing (NLP) tasks. Experiments on the EmoBank, a three dimensional sentiment corpus, show that the proposed domain-distilled BERT outperforms the original BERT and other deep learning models in terms of dimensional sentiment score prediction.

***Keywords:*** scarce resource, domain distillation, sentiment analysis, deep neural network, natural language processing

## 1. INTRODUCTION

In sentiment analysis, affect states can be generally represented using the categorical and dimensional approaches [1,2]. The categorical approach represents affect states using several discrete classes such as positive and negative (binary) or Ekman's [3] six basic emotions (anger, happiness, fear, sadness, disgust, and surprise). Different classification methods can then be used to identify the affective classes. For the dimensional approach, it represents affect states using a continuous numerical value for multiple dimensions such as valence-arousal (VA) space [4], as shown in Fig. 1. The valence dimension reflects the degree of positive and negative sentiment, and the arousal dimension reflects the degree of

Received September 15, 2021; accepted November 30, 2021.
Communicated by Berlin Chen.
+ Corresponding author.
* This work was supported by the Ministry of Science and Technology, Taiwan, under Grant Nos. MOST 107-2628-E-155-002-MY3 and MOST 110-2628-E-155-002.

calm and excitement. Any sentiment expressions can then be represented as a point in the VA coordinate plane, and their VA ratings can be recognized using different regression methods. Compared to the categorical approach, the dimensional approach can provide more fine-grained (real-valued) sentiment analysis, and thus has received considerable attention in recent years [5–20].

Several dimensional sentiment corpora have been proposed in recent years. For example, CVAT [21] is a two-dimensional corpus of 2,969 sentences annotated with VA ratings, consisting of six domains including book review, laptop review, hotel review, car forum, politics forum and news articles. EmoBank [22] is a three-dimensional corpus of 10,062 sentences annotated with valence-arousal-dominance (VAD) ratings, consisting of six domains including fiction, blogs, essays, letters, travel and news articles. Traditional methods trained on such multi-domain corpora typically use a strategy to learn domain-specific features to cover as many domains as possible. Given a rich data resource, as shown in Fig. 2 (a), such a strategy may work well and achieve good prediction performance because the training samples (red point) could cover the domains more completely. However, under conditions of data scarcity, as shown in Fig. 2 (b), this strategy becomes more challenging because sample insufficiency may lead to a smaller coverage of the domain feature space (twill part), causing a further drop in performance.

To tackle the data scarcity problem, this study proposes a domain-distilled method to learn domain-invariant features instead of domain-specific features. As shown in Fig. 3 (a), traditional methods that use a strategy to learn domain-specific features may produce inaccurate prediction results for the test samples (blue star) outside the learned feature space (twill part). Conversely, as shown in Fig. 3 (b), the proposed method distills domain-invariant features from different domains such that both the training (red point) and test samples (blue star) can be transformed into the learned domain-invariant feature space, thus improving prediction performance. The distillation process is accomplished using a domain discriminator to distinguish the feature's domain. In addition, the domain discriminator is trained by maximizing (instead of minimizing) the prediction loss

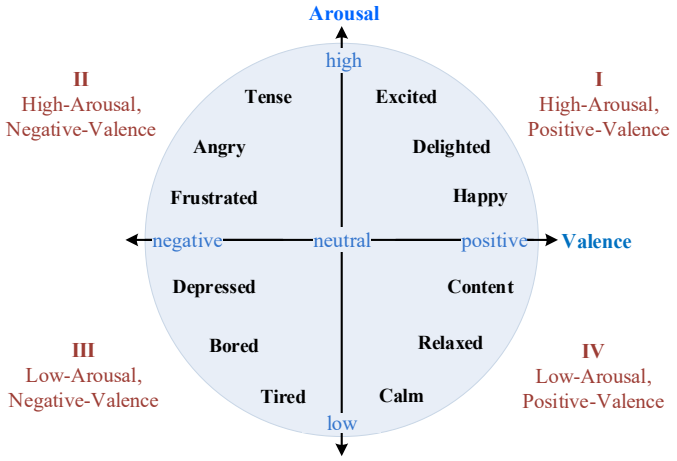
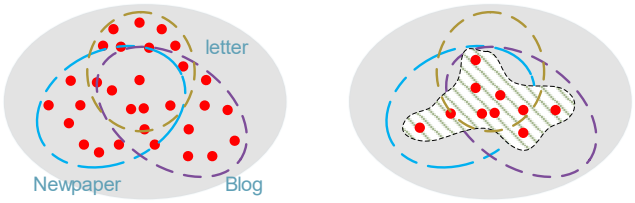
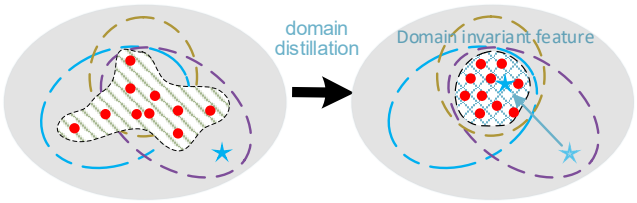



Fig. 1. Dimensional approach to affect state representation.




(a) Rich resource. (b) Scarce resource.

Fig. 2. Conceptual diagram of learning domain-specific features under condition of rich and scarce data resources.




(a) Domain-specific feature. (b) Domain-invariant feature.

Fig. 3. Conceptual diagram of domain-specific and domain-invariant feature learning.

because this makes it difficult for the discriminator to distinguish between domains and thus improving its ability to learn the domain-invariant features, whereas minimizing the prediction loss tends to yield learning of domain-specific features.

To evaluate the proposed method, we implement the domain-distilled method using the Bidirectional Encoder Representations from Transformers (BERT) [23] as the base classifier due to its promising results in many natural language processing (NLP) tasks. The dataset used for evaluation is EmoBank [22], and the goal is to predict the VAD ratings for each sentence. Experimental results show that the proposed domain-distilled BERT outperforms the original BERT and other deep learning models under conditions of data scarcity. For more detailed analysis, we also use a $t$-Distributed Stochastic Neighbor Embedding ($t$-SNE) dimensional reduction technique to visualize the difference of the learned feature space before and after domain distillation.

Our contributions are summarized as follows:

1. We propose a domain-distilled method to learn domain invariant features to improve dimensional sentiment analysis under conditions of data scarcity.

2. We implement the domain-distilled method in BERT to make BERT more adaptable to data scarce conditions.

3. The proposed domain-distilled BERT outperforms the original BERT and other deep learning models on the EmoBank with three dimensions.

The rest of this paper is organized as follows. Section 2 briefly reviews the literature on dimensional sentiment analysis and pre-trained language models. Section 3 presents the proposed domain-distilled BERT. Section 4 summarizes the comparative results of different methods for VAD prediction. Conclusions are finally drawn in Section 5.

## 2. RELATED WORK

This section presents a review of the literature in dimensional sentiment analysis [5–8, 10–20, 24], and pre-trained language models [23, 25–32] .

### 2.1 Dimensional Sentiment Analysis

Recent studies on dimensional sentiment analysis can be categorized as regression-based [5, 6, 13–16] and neural-network-based models [7, 8, 10–12, 17–20, 33].

Regression-based methods have been intensively studied for dimension score prediction. Wei *et al*. [5] proposed a cross-lingual approach that trained a linear regression model using the dimension scores of a set of English seed words (source) and their translated Chinese seed words (target). This was extended by Wang *et al*. [8] using a locally weighted linear regression model. Malandrakis *et al*. [10] built a linear regression model using $n$-grams with sentiment scores as features. Both Paltoglou and Thelwall [9] and Amir *et al*. [6] used support vector regression (SVR). Wang *et al*. [7] developed a community-based weighted graph model that performed the regression task on a graph using a social networking method to predict word dimension scores.

Recently, deep neural network models with word embeddings [34–37] or sentiment embeddings [38–41] have been widely applied to dimensional score prediction. Du and Zhang [12] used a boosted neural network trained on character-enhanced word embeddings to predict word dimension scores. Vilares *et al*. [13] used a CNN trained on Twitter word embeddings to determine the sentiment of tweets from highly negative to highly positive using a five-point scale. Wu *et al*. [14] introduced a densely connected deep LSTM model to concatenate features at different levels to predict the dimension scores of both words and phrases. Goel *et al*. [15] presented an ensemble of different neural networks to determine the intensity level for different emotion categories such as anger, fear, joy and sadness. Zhu *et al*. [20] presented an adversarial attention network to predict the dimension scores of short texts. Yu *et al*. [16] proposed a pipelined neural network model to sequentially learn word intensity and modifier weights for phrase-level sentiment intensity prediction. Wang *et al*. [19] developed a regional CNN-LSTM model that integrates both local (regional) information within sentences and long-distance dependencies across sentences to predict the dimension scores of long texts. They also proposed a capsule tree-LSTM model by introducing a dynamic routing algorithm to improve the performance of tree-LSTM models [42]. Huang *et al*. [17] incorporated a context-dependent sentiment lexicon into a 3-channel CNN to predict the strength of both words and texts. Xie *et al*. [24] presented a multi-dimensional relation model to incorporate relations between dimensions into deep neural networks for dimension score prediction.

All of the above methods were developed with training data sets several times larger than the testing data sets. None of them focuses on sentiment score prediction under data scarcity conditions.

### 2.2 Pre-trained Language Models

Leveraging its ability to extract knowledge from unlabeled data, Google launched BERT [23], and pre-trained language models have since achieved promising results in multiple NLP tasks. These models are pre-trained on large-scale corpora to obtain general

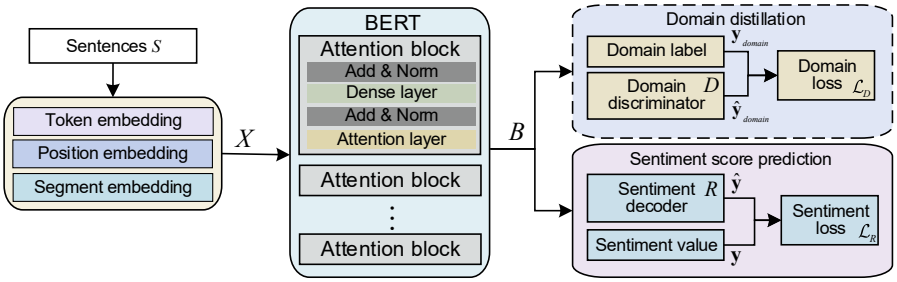



Fig. 4. Overview of the proposed domain-distilled BERT for dimensional sentiment analysis.

language representations and then fine-tuned in specific downstream tasks, such as text generation, question answering and sentiment analysis. Pre-trained language models can be divided into two categories according to their pretraining methods: auto-regression language models [25,27,43,44] and auto-encoder language models [23,26]. Auto-regression language models try to capture language features from the beginning of a sentence to its end. However, autoencoder language models try to capture bi-directional features using masked pre-trained methods, which is very similar to a cloze test. Given a sentence $x = w_1, \ldots, w_i$, with $i$ words, autoregression language models try to learn one direction feature by predicting $w_i$ with $w_1, \ldots, w_{i-1}$. Autoencoder language models try to predict an original sequence $x$ with a masked sequence $w_1, \ldots, t, w_i$, where $t$ represents the masked token [MASK]. Both autoencoder and autoregression language models require fine-tuning to adapt to downstream tasks. First, the ML layer is adapted on the top of pre-trained language models. Then, the pre-trained model and the ML layer are fine-tuned together with a low learning rate to avoiding degradation. The authors in [45] show that pre-trained methods help models achieve more robust results.

From the perspective of neural network structures, differences between pre-trained language models become more subtle. All pre-trained models use the attention mechanism, and some models even share identical neural network structures. For example, BERT [23] and ELECTRA [26] share an identical network architecture, but they use different pre-trained methods. ELECTRA uses generated words to replace masked tokens, avoiding inconsistent inputs in the pre-training and fine-tuning stages. Pre-trained language models use large amounts of unlabeled data to improve downstream task performance, but require labeled data in the fine-tuning stage.

## 3. DOMAIN-DISTILLED BERT

The proposed method consists of three parts: BERT, domain distillation and sentiment score prediction. Fig. 4 shows the overall architecture. Given a sentence, the BERT is first fine-tuned to extract domain-specific features by minimizing the sentiment loss. The domain distillation is then implemented as an add-on module (as shown in the dashed rectangle) into BERT to extract domain-invariant features by maximizing the domain loss. Finally, the extracted domain-invariant features are used to predict the VAD scores of each sentence. The details of BERT and the domain distillation process are described as follows.

### 3.1 BERT

BERT uses Transformer's [46, 47] encoder as its neural network structure. Given a sentence $S$, words are separated into word pieces [48]. For example, 'training' is separated into 'train' and '##ing'. Then, a classification token ([cls]) is manually added at the beginning of the sentence, and punctuation is replaced with a sentence separation token ([sep]). Finally, one-hot encoding, word pieces and special tokens are embedded into token embedding:

$$T = Tokenize(S), \tag{1}$$

where $S$ denotes an input sentence and $T$ denotes the tokenized sentence. In segment embedding, words in different sentences are encoded into different values. The encoding process aims to represent information related to the sentence's order. For position embedding, instead of using the position function to represent word order information, BERT uses learnable position embedding to represent the position information, thus allowing attention networks to capture position information. BERT gathers word, sentence order and position order information by multiplying position embedding with the sum of the token embedding and segment embedding:

$$X = E_{pos}(onehot(T) + E_{seg}), \tag{2}$$

where $E_{pos}$ represents the learned position embedding, $E_{seg}$ represents the segment embedding, $X \in \mathbb{R}^{N \times H}$, $H$ represents the dimensionality of word embedding, $N$ represents the max length of the input sentence, and *onehot* represents the one hot encoding.

Then matrix $X$ is passed through multiple attention blocks, each of which contains a multi-head self-attention layer, a dense layer, and two normalization layers. The self-attention layer with input matrix $X$ is defined as

$$\text{Att}(X) = softmax(\frac{XW_Q W_K^T X^T}{\sqrt{d}})XW_V, \tag{3}$$

where $W_Q, W_K, W_V \in \mathbb{R}^{H \times d}$ respectively denote the trainable parameters for the query, key, and value matrixes. Attention network constructs an attention map using query and key matrixes to find attentions between words, and output the weighted value matrix $Att \in \mathbb{R}^{N \times d}$. Then, multiple self-attention layers are concatenated into one vector, and reshaped through a trainable parameter $W^o$:

$$MultiHead(X) = Concat(Att_1, ..., Att_h)W^O, \tag{4}$$

where $W^O \in \mathbb{R}^{hd \times H}$ represents the trainable parameter and $h$ represents the number of self-attention layers, $MultiHead$ represents the multiple self-attention, also known as multihead attention. Then, BERT builds a residual path to avoid gradient disappearance [49], by adding the multi-head attention layer's input and output together, achieving a more stable gradient through batch normalization (BN):

$$M_{att} = BN(\text{MultiHead}(X)) + X, \tag{5}$$

where *BN* represents the batch normalization layer, and $M_{att} \in \mathbb{R}^{N\times H}$ represents the matrices of the normalized multi-head attention layer. Similar to the attentions layer, the dense layer is designed with a residual path followed by a batch normalization layer:

$$M_{dense} = BN(Dense(M_{att})) + M_{att}, \tag{6}$$

where *Dense* represents the dense layer, and $M_{dense} \in \mathbb{R}^{N\times H}$ represents the output matrices. The input and output of both the attention and dense layers are the same size, making it possible to stack layers to form attentions blocks, which can then be stacked into a large network. Basic BERT stacks 12 attentions blocks, and large BERT stacks 24 blocks. We simplify the final output vector of BERT as:

$$\mathscr{B} = BERT(X). \tag{7}$$

To fine-tune BERT to adapt to the dimensional sentiment analysis task, we use a sentiment decoder on the top of attention blocks. The sentiment decoder for the VAD dimensions is designed as a fully connected network with a linear activation function, defined as

$$\hat{y}_{dim} = R_{dim}(\mathscr{B}) = linear(W^R \mathscr{B} + b^R), \tag{8}$$

where dim represents the VAD dimensions, $W^R$ and $b^R$ represent the trainable parameter and its bias, $\hat{y}_{dim}$ represents the predicted VAD scores of an input sentence in the fine-tune process, $\mathscr{B}$ represents the output of BERT, *R* represents the sentiment decoder. The mean square error (MSE) is used as the loss function, defined as

$$\mathscr{L}_R = \sum_{dim\in\{v,a,d\}} \mathscr{L}_r(\hat{\mathbf{y}}_{dim}, \mathbf{y}_{dim}), \tag{9}$$

$$\mathscr{L}_r(\hat{\mathbf{y}}_{dim}, \mathbf{y}_{dim}) = \frac{1}{2m}\sum_{i=1}^{m} \left\| \hat{y}_{dim}^{(i)} - y_{dim}^{(i)} \right\|^2, \tag{10}$$

where $\hat{\mathbf{y}}_{dim} = \{\hat{y}_{dim}^{(1)}, ..., \hat{y}_{dim}^{(m)}\}$ and $\mathbf{y}_{dim} = \{y_{dim}^{(1)}, ..., y_{dim}^{(m)}\}$ respectively denote the predicted and actual VAD scores of training sentences. After fine-tuning, we obtain domain-specific feature $\mathscr{B}_{\theta_F}$ extracted from BERT with parameters $\theta_F$, and the output of the original BERT in Eq. (7) can be rewritten as

$$\mathscr{B}_{\theta_F} = BERT(X; \theta_F). \tag{11}$$

### 3.2 Domain Distillation

The domain distillation process uses a domain discriminator to transform BERT's output from domain-specific features into domain-invariant features. The domain discriminator is designed as a two-layer fully connected network, defined as

$$\hat{y}_{domain}^{c} = D(\mathscr{B}_{\theta_F}) = \text{SoftMax}(W^D{}_{\theta_F} + b^D), \tag{12}$$

where *D* represents the domain discriminator, $W^D$ and $b^D$ represent the trainable parameter and its bias, $\hat{y}_{domain}^{c}$ represents the prediction probability, and $SoftMax$ denotes the

output layer's activation function. Moreover, we minimize the loss function $\mathscr{L}_D$ to train the discriminator $D$:

$$\mathscr{L}_D(\hat{\mathbf{y}}_{domain}, \mathbf{y}_{domain}) = -\sum_c \hat{y}^c_{domain} \log y^c_{domain} \tag{13}$$

where $\hat{y}^c_{domain}$ and $y^c_{domain}$ respectively represent the true and prediction probability of domain label. In the discriminator training process, we obtain the parameter $\theta_D$ for the discriminator $D$. After discriminator training, we freeze $\theta_D$ and adjust BERT to fool the discriminator using the loss function defined as

$$\mathscr{L} = \mathscr{L}_R - \lambda \mathscr{L}_D, \tag{14}$$

where $\lambda$ controls the trade-off between the sentiment loss and domain loss, respectively defined in Eqs. (9) and (12). After the domain distillation, the domain-invariant features can be extracted from BERT with parameters $\tilde{\theta}_F$, and the output of the fine-tuned BERT in Eq. (11) can be rewritten as

$$\mathscr{B}_{\tilde{\theta}_F} = BERT(X; \tilde{\theta}_F). \tag{15}$$

### 3.3 Sentiment Score Prediction

The domain-invariant features $\mathscr{B}_{\tilde{\theta}_F}$ learned in the domain distillation process are then used to predict the final VAD scores, defined as

$$\tilde{y}_{dim} = R_{dim}(\mathscr{B}_{\tilde{\theta}_F}; \tilde{\theta}_{R,dim}), \tag{16}$$

where $\tilde{y}_{dim}$ represents the final predicted VAD scores of an input sentence, $R$ represents the sentiment decoder defined in Eq. (8), and represents the parameter of the sentiment decoder for the VAD dimensions.

## 4. EXPERIMENTS

This section presents the comparative results of the proposed domain-distilled BERT against original BERT and other deep learning models for dimensional sentiment analysis. A series of $t$-SNE visualizations is also used to show the difference between the features learned before and after domain distillation.

### 4.1 Experimental Settings

**Dataset** Experiments are conducted using the EmoBank [22] containing 10,062 sentences with VAD scores. There are six domains in the EmoBank and their distribution is presented in Table 1.

**Training and Testing Partition** To simulate rich and scarce data conditions, we respectively use the standard $k$-fold and reverse $k$-fold cross-validation for evaluation. The standard $k$-fold cross-validation uses $k$–1 folds for training and the remaining one fold for testing, whereas the reverse $k$-fold cross-validation uses one fold for training and the remaining $k$–1 folds for testing.

**Evaluation Metric** Performance is evaluated using Pearson's correlation coefficient ($r$) and Spearman's rank correlation coefficient ($\rho$). Pearson's correlation coefficient is defined as

$$r = \frac{1}{n-1}\sum_{i=1}^{n}\left(\frac{\hat{Y}_i - \bar{\hat{Y}}}{\sigma_{\hat{Y}}}\right)\left(\frac{Y_i - \bar{Y}}{\sigma_Y}\right), \tag{17}$$

where $A_i$ is the actual value, $P_i$ is the predicted value, $n$ is the number of test samples, $\overline{A}$ and $P$ respectively denote the arithmetic mean of $A$ and $P$, and $\sigma$ is the standard deviation. Spearman's rank correlation coefficient is defined as

$$\rho = \frac{\sum_i (\hat{Y}_i - \bar{\hat{Y}})(Y_i - \bar{Y})}{\sqrt{\sum_i (\hat{Y}_i - \bar{\hat{Y}})^2 \sum_i (Y_i - \bar{Y})^2}}. \tag{18}$$

### 4.2 Implementation Details

The implementation details of different methods used for VAD prediction are described as follows.

- **CNN** [13] provides a standard architecture which consists of both the convolution and the pooling layers to map variable-length sentences or texts into fixed-size distributed representations to extract active local $n$-gram features.
- **LSTM** [11] sequentially represents a sentence or text with word order information to determine long-distance dependencies that could help capture the sentiments of long texts.
- **2-layer Bi-LSTM** [50] The standard LSTM model can be enhanced by introducing a bi-directional strategy and stacking multiple layers to form a hierarchical representation.
- **CNN-LSTM** [19] The CNN and LSTM can be combined to form a hierarchical representation by stacking an LSTM layer on top of CNN (CNN-LSTM). This model can simultaneously leverage both local and long-distance features within the sentences.
- **BERT** [23] We use Bert4Keras to implement BERT in our experiments. Bert4Keras is a light re-implementation of the transformer models in Keras' version. In our experiment, we use the BERT-Base model for evaluation. For fine-tuning, we use multilingual cased pre-trained models, pre-trained by Google. BERT was fine-tuned in 4 epochs with batch sizes of 128, using the Adam [51] optimization scheme with a learning rate of 2e-5.
- **DT-BERT (domain-distilled BERT)** In the fine-tuning process, the hyper-parameters of the learning rate, batch size and fine-tuning epochs are identical to those of BERT. We train the discriminator for 50 epochs, and use a feature extractor to fool the discriminator for 2 epochs with = 0.1. The trade-off coefficient $\lambda$ between sentiment loss and domain classification loss is set to 0.1.

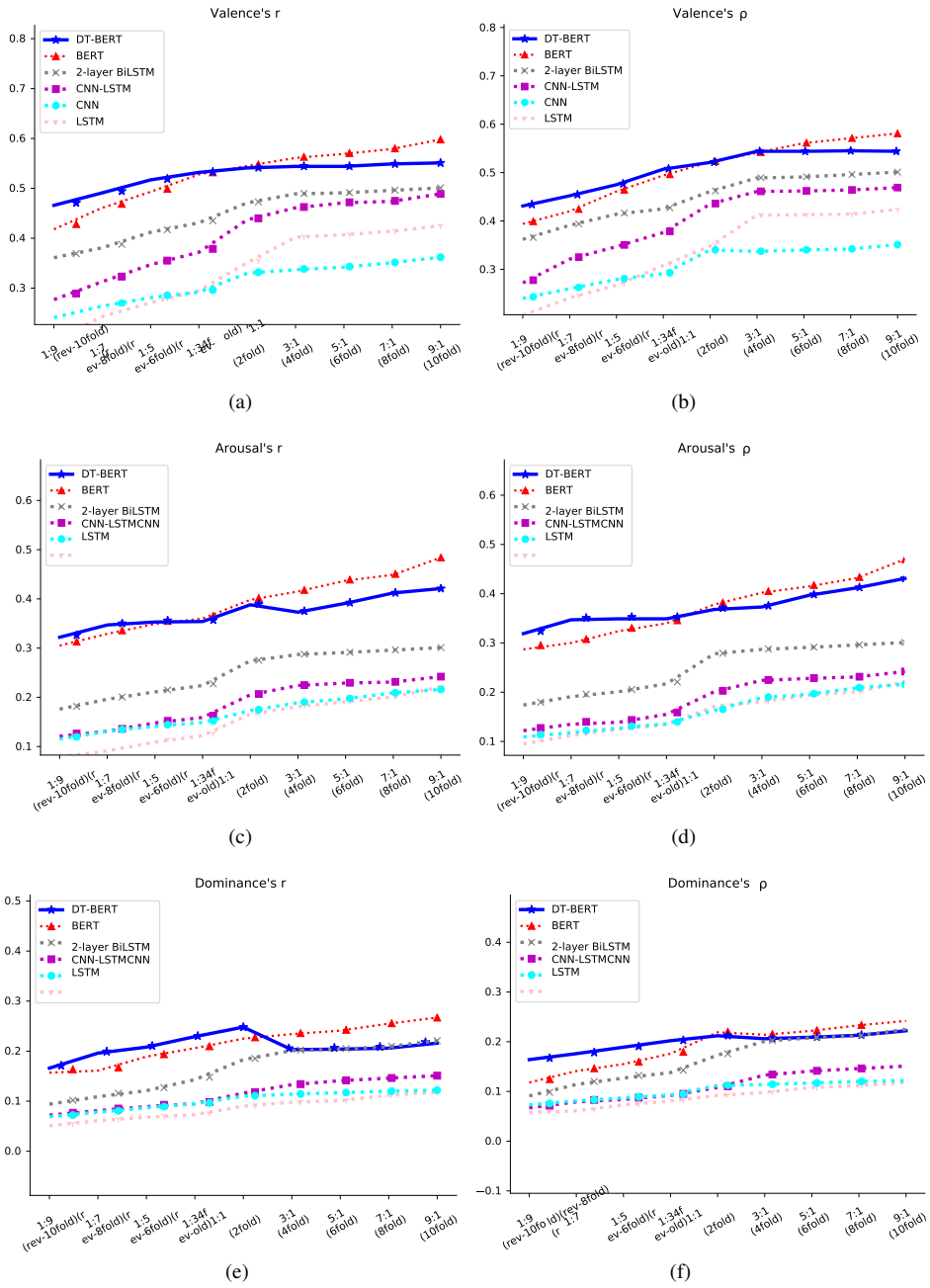
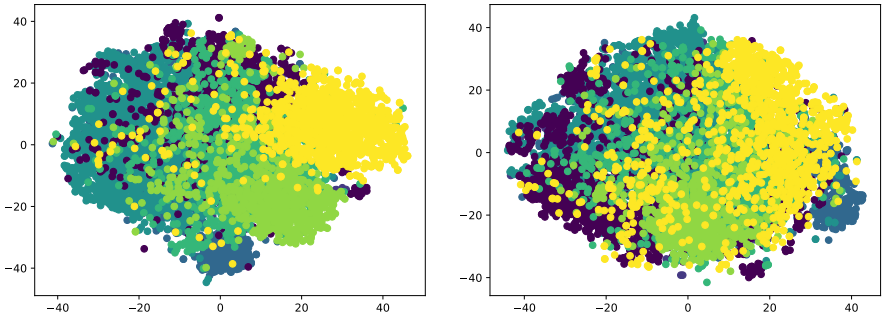



Fig. 5. Result of different methods for VAD prediction under rich and scarce data conditions.

### 4.3 Comparative Result

Fig. 5 shows the performance of the different methods for VAD prediction under rich and scarce data conditions. The $x$-axis represents the split ratio of training and testing data from 1:9 (reverse 10-fold cross validation) to 9:1 (standard 10-fold cross validation). The $y$-axis represents $r$ and $\rho$. The results show that, as training data increases from scarce data to rich data, the prediction performance of all methods also increased. In addition, the transformer-based methods (BERT and DT-BERT) outperformed both hierarchical (2-layer BiLSTM and CNN-LSTM) and single-layer (CNN and LSTM) neural network models for all dimensions. The proposed DT-BERT outperformed BERT under scarce data condition (1:9, 1:7, 1:5 and 1:3), demonstrating that learning domain-invariant features can overcome the data scarce problem. Once the split ratio exceeded 1:1, BERT achieved better performance because the increased volume of training data allows for more domain-specific features to be learned.

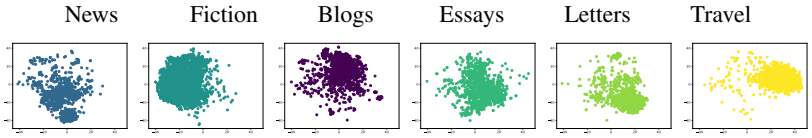
For detailed analysis, we compare the results of BERT and DT-BERT for different domains under the scarce data condition (reverse 10-fold cross validation), as shown in Table 1. The results show that the DT-BERT outperformed BERT for all domains. For the six domains, both methods achieved lowest performance in the travel domain, possibly due to its having the smallest amount of data. For the three dimensions, the valence dimension outperforms arousal, which in turn outperforms dominance, indicating that the dominance dimension is relatively difficult to predict.



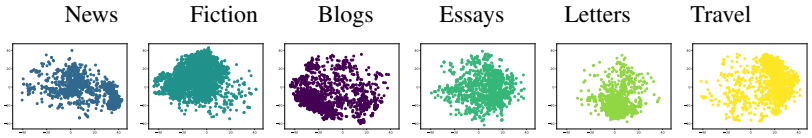
(a) BERT – Overall feature distribution.



(b) DT-BERT – Overall feature distribution.




(c) BERT – Feature distribution of different domains.




(d) DT-BERT – Feature distribution of different domains.

Fig. 6. The $t$-SNE visualization of feature distribution learned by BERT and DT-BERT.

**Table 1. Results of BERT and DT-BERT for different domains under the scarce data condition (reverse 10-fold cross validation).**

| Domain | Num. (%) | Method | Valence | | Arousal | | Dominance | |
|---|---|---|---|---|---|---|---|---|
| | | | $r$ | $\rho$ | $r$ | $\rho$ | $r$ | $\rho$ |
| News | 2,506 (25%) | BERT | 0.430 | 0.421 | 0.335 | 0.284 | 0.136 | 0.144 |
| | | DT-BERT | 0.474 | 0.452 | 0.358 | 0.321 | 0.179 | 0.161 |
| Fiction | 2,753 (28%) | BERT | 0.376 | 0.365 | 0.347 | 0.304 | 0.111 | 0.091 |
| | | DT-BERT | 0.432 | 0.384 | 0.372 | 0.340 | 0.146 | 0.129 |
| Blogs | 1,336 (13%) | BERT | 0.314 | 0.310 | 0.223 | 0.160 | 0.091 | 0.072 |
| | | DT-BERT | 0.371 | 0.336 | 0.284 | 0.227 | 0.120 | 0.110 |
| Essays | 1,135 (11%) | BERT | 0.285 | 0.296 | 0.319 | 0.256 | 0.143 | 0.135 |
| | | DT-BERT | 0.354 | 0.336 | 0.346 | 0.281 | 0.191 | 0.162 |
| Letters | 1,413 (14%) | BERT | 0.316 | 0.301 | 0.165 | 0.142 | 0.096 | 0.081 |
| | | DT-BERT | 0.353 | 0.340 | 0.186 | 0.164 | 0.131 | 0.147 |
| Travel | 919 (9%) | BERT | 0.244 | 0.232 | 0.091 | 0.080 | 0.034 | 0.005 |
| | | DT-BERT | 0.292 | 0.259 | 0.158 | 0.116 | 0.054 | 0.067 |
| Total | 10,062 (100%) | BERT | 0.421 | 0.394 | 0.301 | 0.287 | 0.154 | 0.118 |
| | | DT-BERT | 0.471 | 0.431 | 0.341 | 0.319 | 0.181 | 0.164 |

### 4.4 Visualization of Feature Distribution

To compare the domain-specific features used in BERT and domain-invariant features used in the proposed method, we used a series of $t$-SNE visualizations to show the difference between the features learned before and after domain distillation, as shown in Fig. 6. These visualizations were produced by analyzing the output vectors of BERT and DT-BERT, and each color represents one of the six domains. Comparing Figs. 6 (a) and (b) shows that the features of different domains learned by BERT tend to be distributed in certain locations because the domain information was emphasized, whereas the feature distribution of the DT-BERT is more diverse because the domain-invariant features contain less domain information. Figs. 6 (c) and (d) also show similar results for different domains.

## 5. CONCLUSIONS

This study proposes the domain-distilled BERT for dimensional sentiment analysis under conditions of data scarcity. Compared to traditional methods that use domain-specific features, the proposed domain-distilled method can learn domain-invariant features to improve prediction performance. In addition, the proposed method implemented

on BERT also makes BERT more adaptable to data scarce conditions.Experiments on the EmoBank with VAD dimensions show that the domain-distilled BERT outperforms the original BERT and other deep learning models. The *t*-SNE visualizations also show the difference between the feature distribution learned before and after domain distillation. Future work will focus on applying the proposed method to other possible dimensions and downstream applications. Another direction is to investigate other methods to learn domain-invariant features to improve prediction performance.

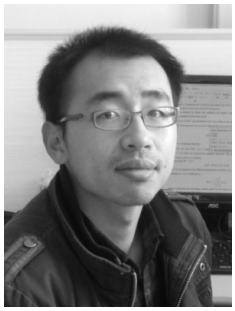
## REFERENCES

1. R. Calvo and S. Kim, "Emotions in text: Dimensional and categorical models," *Computational Intelligence*, Vol. 29, 2013, pp. 527-543.
2. H. Gunes and B. Schuller, "Categorical and dimensional affect analysis in continuous input: Current trends and future directions," *Image and Vision Computing*, Vol. 31, 2013, pp. 120-136.
3. P. Ekman, "An argument for basic emotions," *Cognition & Emotion*, Vol. 6, 1992, pp. 169-200.
4. A. Mehrabian and J. A. Russell, "The basic emotional impact of environments," *Perceptual and Motor Skills*, Vol. 38, 1974, pp. 283-301.
5. W. L. Wei, C. H. Wu, and J. C. Lin, "A regression approach to affective rating of chinese words from anew," in *Proceedings of the 4th International Conference on Affective Computing and Intelligent Interaction*, 2011, pp. 121-131.
6. S. Amir, B. C. Wallace, H. Lyu, P. Carvalho, and M. J. Silva, "Modelling context with user embeddings for sarcasm detection in social media," in *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning*, 2016, pp. 167-177.
7. J. Wang, L. C. Yu, K. R. Lai, and X. Zhang, "Community-based weighted graph model for valence-arousal prediction of affective words," *IEEE/ACM Transactions on Audio Speech and Language Processing*, Vol. 24, 2016, pp. 1957-1968.
8. J. Wang, L. C. Yu, K. R. Lai, and X. Zhang, "Locally weighted linear regression for cross-lingual valence-arousal prediction of affective words," *Neurocomputing*, Vol. 194, 2016, pp. 271-278.
9. G. Paltoglou and M. Thelwall, "Seeing stars of valence and arousal in blog posts," *IEEE Transactions on Affective Computing*, Vol. 4, 2013, pp. 116-123.
10. N. Malandrakis, A. Potamianos, E. Iosif, and S. Narayanan, "Distributional semantic models for affective text analysis," *IEEE Transactions on Audio, Speech, and Language Processing*, Vol. 21, 2013, pp. 2379-2392.
11. K. S. Tai, R. Socher, and C. D. Manning, "Improved semantic representations from tree-structured long short-term memory networks," in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing*, Vol. 1, 2015, pp. 1556-1566.
12. S. Du and X. Zhang, "Aicyber's system for ialp 2016 shared task: Character-enhanced word vectors and boosted neural networks," in *Proceedings of International Conference on Asian Language Processing*, 2016, pp. 161-163.

13. D. Vilaresa, Y. Doval, M. A. Alonsoa, and C. Gómez-Rodríguez, "Lys at semeval-2016 task 4: Exploiting neural activation values for twitter sentiment classification and quantification," in *Proceedings of the 10th International Workshop on Semantic Evaluation*, 2016, pp. 79-84.
14. C. Wu, F. Wu, Y. Huang, S. Wu, and Z. Yuan, "THU NGN at IJCNLP-2017 Task 2: Dimensional sentiment analysis for Chinese phrases with deep lstm," in *Proceedings of the 8th International Joint Conference on Natural Language Processing*, 2017, pp. 47-52.
15. P. Goel, D. Kulshreshtha, P. Jain, and K. K. Shukla, "Prayas at emoint 2017: An ensemble of deep neural architectures for emotion intensity prediction in tweets," in *Proceedings of the 8th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, 2017, pp. 58-65.
16. L. C. Yu, J. Wang, K. R. Lai, and X. Zhang, "Pipelined neural networks for phrase-level sentiment intensity prediction," *IEEE Transactions on Affective Computing*, Vol. 11, 2020, pp. 447-458.
17. M. Huang, H. Xie, Y. Rao, J. Feng, and F. L. Wang, "Sentiment strength detection with a context-dependent lexicon-based convolutional neural network," *Information Sciences*, Vol. 520, 2020, pp. 389-399.
18. S. Akhtar, D. Ghosal, A. Ekbal, P. Bhattacharyya, and S. Kurohashi, "All-in-one: Emotion, sentiment and intensity prediction using a multi-task ensemble framework," *IEEE Transactions on Affective Computing*, Vol. 3045, 2019, p. 1.
19. J. Wang, L. C. Yu, K. R. Lai, and X. Zhang, "Tree-structured regional cnn-lstm model for dimensional sentiment analysis," *IEEE/ACM Transactions on Audio Speech and Language Processing*, Vol. 28, 2020, pp. 581-591.
20. S. Zhu, S. Li, and G. Zhou, "Adversarial attention modeling for multi-dimensional emotion regression," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019, pp. 471-480.
21. L. C. Yu, L. H. Lee, S. Hao, J. Wang, Y. He, J. Hu, K. R. Lai, and X. Zhang, "Building Chinese affective resources in valence-arousal dimensions," in *Proceedings of the 15th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2016, pp. 540-545.
22. S. Buechel and U. Hahn, "Emobank: Studying the impact of annotation perspective and representation format on dimensional emotion analysis," in *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, Vol. 2, 2017, pp. 578-585.
23. J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of Conference of the North American Chapter of the Association for Computational Linguistics*, Vol. 1, 2019, pp. 4171-4186.
24. H. Xie, W. Lin, S. Lin, J. Wang, and L. C. Yu, "A multi-dimensional relation model for dimensional sentiment analysis," *Information Sciences*, Vol. 579, 2021, pp. 832-844.
25. Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov, and Q. V. Le, "Xlnet: Generalized autoregressive pretraining for language understanding," *Advances in Neural Information Processing Systems*, 2019, pp. 5754-5764.

26. C. Kevin, L. Minh-Thang, V. L. Quoc, and D. Christopher, "Electra : Pre-training text encoders as discriminators rather than generators," in *Proceedings of the 8th International Conference on Learning Representations*, 2020, pp. 1-18.
27. A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," *OpenAI Blog*, Vol. 1, 2018, p. 9.
28. C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *arXiv Preprint*, 2019, arXiv:1910.10683.
29. W. Liu, P. Zhou, Z. Zhao, Z. Wang, Q. Ju, H. Deng, and P. Wang, "K-BERT: Enabling language representation with knowledge graph," in *Proceedings of AAAI Conference on Artificial Intelligence*, 2020, pp. 2901-2908.
30. M. Zaheer, G. Guruganesh, A. Dubey, J. Ainslie, C. Alberti, S. Ontanon, P. Pham, A. Ravula, Q. Wang, L. Yang, and A. Ahmed, "Big bird: Transformers for longer sequences," in *Proceedings of the 34th Conference on Neural Information Processing Systems*, 2020, pp. 1-51.
31. Y. Cui, W. Che, T. Liu, B. Qin, S. Wang, and G. Hu, "Revisiting pre-trained models for Chinese natural language processing," in *Findings of the Association for Computational Linguistics: EMNLP*, 2020, pp. 657-668.
32. Y. Zhang, J. Wang, L. C. Yu, and X. Zhang, "MA-BERT: Learning representation by incorporating multi-attribute knowledge in transformers," in *Findings of the Association for Computational Linguistics: ACL-IJCNLP*, 2021, pp. 2338-2343.
33. J. Wang, L. C. Yu, K. R. Lai, and X. Zhang, "Dimensional sentiment analysis using a regional cnn-lstm model," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, 2016, pp. 225-230.
34. T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proceedings of Advances in Neural Information Processing Systems*, 2013, pp. 1-9.
35. T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," in *Proceedings of the 1st International Conference on Learning Representations*, Vol. 2013, 2013, pp. 1-12.
36. J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Proceedings of Conference on Empirical Methods in Natural Language Processing*, 2014, pp. 1532-1543.
37. P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *Transactions of the Association for Computational Linguistics*, Vol. 5, 2017, pp. 135-146.
38. D. Tang, F. Wei, B. Qin, N. Yang, T. Liu, and M. Zhou, "Sentiment embeddings with applications to sentiment analysis," *IEEE Transactions on Knowledge and Data Engineering*, Vol. 28, 2016, pp. 496-509.
39. L. C. Yu, J. Wang, K. R. Lai, and X. Zhang, "Refining word embeddings for sentiment analysis," in *Proceedings of Conference on Empirical Methods in Natural Language Processing*, 2017, pp. 534-539.
40. L. C. Yu, J. Wang, K. R. Lai, and X. Zhang, "Refining word embeddings using intensity scores for sentiment analysis," *IEEE/ACM Transactions on Audio Speech and Language Processing*, Vol. 26, 2018, pp. 671-681.

41. J. Wang, Y. Zhang, L. C. Yu, and X. Zhang, "Contextual sentiment embeddings via bidirectional gru language model," *Knowledge-Based Systems*, Vol. 235, 2022, Article 107663.
42. J. Wang, L. C. Yu, K. Robert Lai, and X. Zhang, "Investigating dynamic routing in tree-structured lstm for sentiment analysis," in *Proceedings of Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, 2019, pp. 3432-3437.
43. T. B. Brown, B. Mann, N. Ryder *et al.*, "Language models are few-shot learners," *arXiv Preprint*, 2020, arXiv:2005.14165.
44. W. Fedus, B. Zoph, and N. Shazeer, "Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity," *arXiv Preprint*, 2021, arXiv:2101.03961.
45. D. Jin, Z. Jin, J. T. Zhou, and P. Szolovits, "Is BERT really robust? A strong baseline for natural language attack on text classification and entailment," in *Proceedings of AAAI Conference on Artificial Intelligence*, 2020, pp. 8018-8025.
46. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in Neural Information Processing Systems*, 2017, pp. 5999-6009.
47. U. Naseem, I. Razzak, K. Musial, and M. Imran, "Transformer based deep intelligent contextual embedding for twitter sentiment analysis," *Future Generation Computer Systems*, Vol. 113, 2020, pp. 58-69.
48. S. F. Huckemann and B. Eltzner, "Backward nested descriptors asymptotics with inference on stem cell differentiation," *Annals of Statistics*, Vol. 46, 2018, pp. 1994-2019.
49. K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770-778.
50. A. Graves, N. Jaitly, and A. R. Mohamed, "Hybrid speech recognition with deep bidirectional lstm," in *Proceedings of IEEE Workshop on Automatic Speech Recognition & Understanding*, 2013, pp. 273-278.
51. D. P. Kingma and J. Ba, "Adam: a method for stochastic optimization," in *Proceedings of International Conference on Learning Representation*, 2015, pp. 1-15.



**Wei Lin** received the M.Eng. degree with the College of Mathematics and Computer Science in Fuzhou University. He is currently pursuing Ph.D. degree with the Department of Information Management, Yuan Ze University, Taoyuan, Taiwan. His research interests include natural language processing, text mining, and machine learning.

**Liang-Chih Yu** received the Ph.D. degree in Computer Science and Information Engineering from National Cheng Kung University, Tainan, Taiwan. He is a Professor with the Department of Information Management, Yuan Ze University, Taoyuan City, Taiwan. He was a Visiting Scholar with the Natural Language Group, Information Sciences Institute, University of Southern California, from 2007 to 2008, and with DOCOMO Innovations for three months in 2018. His research interests include natural language processing, sentiment analysis, computer-assisted language learning. He is currently Board Member and Convener of SIGCALL of the Association for Computational Linguistics and Chinese Language Processing, and is an Editorial Board Member of International Journal of Computational Linguistics and Chinese Language Processing. His team has developed systems that ranked first in IJCNLP 2017 Task 4: Customer Feedback Analysis, and second in the recent SemEval and BEA shared task competitions.