Structured Probabilistic Coding

Dou Hu[1,2], Lingwei Wei[1], Yaxin Liu[1,2], Wei Zhou[1*], Songlin Hu[1,2*]

[1] Institute of Information Engineering, Chinese Academy of Sciences
[2] School of Cyber Security, University of Chinese Academy of Sciences
{hudou, weilingwei, liuyaxin, zhouwei, husonglin}@iie.ac.cn

Abstract

This paper presents a new supervised representation learning framework, namely structured probabilistic coding (SPC), to learn compact and informative representations from input related to the target task. SPC is an encoder-only probabilistic coding technology with a structured regularization from the target space. It can enhance the generalization ability of pre-trained language models for better language understanding. Specifically, our probabilistic coding simultaneously performs information encoding and task prediction in one module to more fully utilize the effective information from input data. It uses variational inference in the output space to reduce randomness and uncertainty. Besides, to better control the learning process of probabilistic representations, a structured regularization is proposed to promote uniformity across classes in the latent space. With the regularization term, SPC can preserve the Gaussian structure of the latent code and achieve better coverage of the hidden space with class uniformly. Experimental results on 12 natural language understanding tasks demonstrate that our SPC effectively improves the performance of pre-trained language models for classification and regression. Extensive experiments show that SPC can enhance the generalization capability, robustness to label noise, and clustering quality of output representations.

Introduction

Probabilistic embedding (Vilnis and McCallum 2015) is a flexible representation learning technology whose goal is to learn the underlying probability distribution of data. In contrast to deterministic embedding (Pereyra et al. 2017; Miyato, Dai, and Goodfellow 2017; Gunel et al. 2021), which maps each data to a fixed vector representation, probabilistic embedding embraces the notion of learning a probability distribution, mapping each data point to a distribution. These probabilistic embedding approaches can better describe the uncertainty and complexity of data, handle redundant information, and provide better discriminative representations. Probabilistic embedding has been applied to various domains such as computer vision (Oh et al. 2019; Shi and Jain 2019) and natural language processing (Mahabadi, Belinkov, and Henderson 2021; Hu et al. 2022).

Most probabilistic embedding methods (Kingma and Welling 2014; Alemi et al. 2017; Higgins et al. 2017; Fischer 2020; An, Jammalamadaka, and Chong 2023) are (or can be) built upon the information bottleneck (IB) principle (Tishby, Pereira, and Bialek 1999; Tishby and Zaslavsky 2015). The principle aims to find a maximally compressed representation of the input that preserves as much as possible information about the target task, striking a balance between compression and prediction. These IB-based methods typically involve two parametric modules, i.e., an encoder and a decoder (Goldfeld and Polyanskiy 2020). Usually, the encoder maps the input to a probabilistic distribution in the latent space, and the decoder maps the probabilistic distribution to the output representations in the target task space.

However, under the encoder-decoder architecture, the process of mapping input data to probability distributions by the encoder may lose some task-related information, which is essential for the decoder during the learning process. This is because probability distributions inherently contain randomness and uncertainty, which may be irrelevant to the task and interfere with the task prediction process of the decoder. To avoid this, we propose an encoder-only embedding technology, **probabilistic coding**, that integrates probabilistic encoding and task prediction into one module. By using variational inference in the output space, we can better control and utilize randomness and uncertainty of data. The learned compact representations can fully capture the underlying structure of data, and preserve the effective information from input related to target task. This helps improve model generalization performance, especially when facing limited data or noisy labels.

Besides, although probabilistic embedding methods can capture data uncertainty and complexity, they are restricted to limited or biased data, which cannot fully represent the true distribution of the target task. In the process of mapping input data to probability distributions in the latent space by the encoder, some task-related important information may be missing to some extent. The insufficient information lead to poor task performance on new data and inadequate model generalization. To improve task prediction ability of the latent representations, we leverage the structured information of target task space to constrain the learning process of the probability distribution in latent space. Under the framework of probabilistic coding, the **structured regularization** of la-

[*]Corresponding author.

tent space can help the model learn more informative representations related to the target task, thereby improving the model's prediction accuracy on new data.

In this paper, we present a new supervised representation learning framework, **structured probabilistic coding** (SPC), an encoder-only probabilistic coding technology with a structured regularization from the target label space. By extracting compact and informative representations from input related to the target task, SPC can enhance the generalization ability of pre-trained language models for better language understanding. Specifically, the probabilistic coding technique performs variational approximation to encode the input into stochastic output representations under Gaussian distribution spaces, while minimizing the conditional entropy of the target label given the representations. Besides, the structure information of target task space is introduced to constrain the probability distribution of latent space. The structured regularization encourages class-level uniformity within the latent space under the multivariate Gaussian distribution, making the distribution better reflect task-related information, which is beneficial for task prediction. Under the probabilistic coding framework with the regularization term, SPC can maintain the Gaussian structure of the latent code while achieving the best possible coverage of the hidden space with uniformity across classes.

We conduct experiments on 12 natural language understanding tasks, including 10 classification tasks such as emoji prediction, hate speech detection, irony detection, offensive language detection, sentiment analysis, stance detection, emotion detection from different domains, as well as 2 regression tasks including semantic similarity prediction and plausible clarifications ranking. The results demonstrate that our SPC effectively improves the performance of pre-trained language models for classification and regression tasks. For instance, with the RoBERTa backbone, SPC improves average performance by **+3.9%** and **+1.7%** for classification and regression tasks compared to CE/MSE. Our SPC framework consistently achieves the best average performance compared to other methods, including deterministic and probabilistic approaches, under different backbone models such as BERT and RoBERTa. Extensive experiments show that SPC can enhance the model generalization capability including out-of-distribution and data-constrained scenarios, robustness to label noise, and clustering quality of output representations.

The main contributions are as follows: 1) We propose an encoder-only probabilistic coding method that integrates probabilistic encoding and task prediction into one module. It maximally preserves the effective information from input related to target task. 2) We design a structured regularization term to promote class-level uniformity in the latent space for better task prediction ability of probabilistic embedding. 3) We present a supervised representation learning framework named SPC, to learn compact and informative representations from input related to the target task. It can enhance the generalization ability of pre-trained language models for better language understanding. 4) Experiments on 12 benchmarks show that SPC achieves state-of-the-art performance on classification and regression tasks. Extensive experiments reveal that SPC can enhance the generalization capability, robustness to label noise, and clustering quality of output representations.[1]

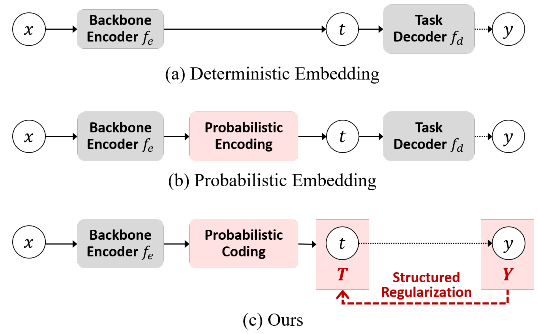

Figure 1: Comparison of our SPC with existing deterministic embedding and probabilistic embedding methods.

Methodology

In this section, we present a supervised representation learning framework, **structured probabilistic coding** (SPC), to learn compact and informative representations from input related to the downstream task. As shown in Figure 1(c), SPC is an encoder-only probabilistic coding technology with a structured regularization from the target label space.

Probabilistic Coding The probabilistic coding technology integrates probabilistic encoding and task prediction into one module. Different from existing probabilistic embeddings applying encoder-decoder architecture, our encoder-only paradigm can effectively maintain task-related features and avoid negative effects caused by the randomness and uncertainty of probabilistic encoding.

Under the Markov chain constraint $Y \to X \to T$, we have $p(t|x, y) = p(t|x)$. we aim to minimize the mutual information between the input X and the latent representation T, as well as maximize the information between the representations T and the target label Y. Specifically, we employ a variational approximation to encode each input x into a Gaussian distribution representation t in the output space $\mathcal{Y}$, i.e., $T \in \mathbb{R}^{|\mathcal{Y}|}$. Additionally, we maximize the lower bound of $I(T;Y)$ by estimating the conditional entropy of the target label Y given the representations T. The objective of probabilistic coding can be:

$$\mathcal{L}_{PC} = \mathbb{E}_{t\sim p_\theta(t|x)}[-\log q(y|t)] + \beta KL(p_\theta(t|x); r(t)), \quad (1)$$

where $q(y|t)$ is a non-parametric operation, e.g., softmax function for classification. $r(t)$ is an estimate of the prior $p(t)$ of t. $p_\theta(t|x)$ is a variational estimate of the posterior probability of t and is learned by the stochastic encoder θ. $KL(\cdot)$ denotes the analytic KL-divergence term, serving as the regularization that forces the posterior probability of t to approximately converge to the prior $p(t)$. $\beta > 0$ is a hyperparameter which controls the trade-off between the sufficiency of t for predicting y and the compression of t from x.

[1]The code is available at https://github.com/zerohd4869/SPC

Let the prior $p(t)$ be the isotropic Gaussian distribution. And let the variational approximate posterior $p_\theta(t|x)$ be a multivariate Gaussian with a diagonal covariance structure, i.e., $p_\theta(t|x) = \mathcal{N}(t; \mu(x), \Sigma(x))$, where μ and Σ represent the mean and diagonal covariance. Both of their parameters are input-dependent and predicted by an MLP (a fully-connected neural network with a single hidden layer), respectively. As the sampling of t is a stochastic process, we apply the re-parameterization trick (Kingma and Welling 2014) to ensure unbiased gradients for the model.

In existing IB-based methods (Alemi et al. 2017; Fischer 2020; An, Jammalamadaka, and Chong 2023) with an encoder-decoder architecture, their decoder can be a parametric approximation q_ϕ of $p(y|t)$. That is, the compressed representation t can be sampled from the distribution $p_\theta(t|x)$, meaning that a specific pattern of noise is added to the input of $q_\phi(y|t)$. The noise could diminish the information conveyed by t and potentially cause a loss of task-related information, which is crucial for the decoder ϕ during the learning process. Different from them, our probabilistic coding integrates probabilistic encoding and task prediction into one encoder module with the network θ, and applies a non-parametric operation to obtain prediction outputs. It effectively avoids negative impacts caused by the randomness and uncertainty of probabilistic encoding.

Structured Regularization As mentioned above, the Markov assumption restricts that the representation T cannot depend directly on the target task Y. This means that the learning of t does not fully utilize information of label space. Accordingly, the learned representation cannot sufficiently represent the true distribution of the target task, leading to poor generalization ability when learning from the limited or biased data. To enhance the prediction ability of the latent representation, we design a new structured regularization related to the target task space. It can constrain the learning process of probability distributions in the latent space.

Specifically, we add an additional regularization term about the latent distribution to the objective function that maximizes the prior entropy of T on the label space:

$$\max H(T), \text{ where } T \in \mathbb{R}^{|\mathcal{Y}|}. \tag{2}$$

and

$$H(T) = -\frac{1}{|\mathcal{X}|}\sum_{i=1}^{|\mathcal{X}|}\sum_{j=1}^{|\mathcal{Y}|} p_{i,j}\log p_{i,j} \tag{3}$$

where $|\mathcal{X}|$ is the number of the expected data. $|\mathcal{Y}|$ is the number of classes. $p_{i,j}$ represents the probability of sample i belonging to class j in the latent space, which is computed by the output of p_θ, i.e., the encoder of probabilistic embedding.

In the implementation, the Jensen's inequality is applied to estimate the upper bound of $H(T)$. Given each sampled batch data, we have:

$$\begin{aligned} H(T_b) \le \hat{H}(T_b) &\approx -\sum_{j=1}^{|\mathcal{Y}|}\left(\frac{1}{B}\sum_{k=1}^{B} p_{k,j}\right)\log\left(\frac{1}{B}\sum_{l=1}^{B} p_{l,j}\right) \\ &\approx -\sum_{j=1}^{|\mathcal{Y}|}\overline{p_{\cdot,j}}\log\left(\overline{p_{\cdot,j}}\right) = \mathcal{L}_b, \end{aligned} \tag{4}$$

where B is the batch size. $\overline{p_{\cdot,j}} = \frac{1}{N}\sum_{i=1}^{N} p_{i,j}, j \in \{1, ..., |\mathcal{Y}|\}$ represents the average predicted probability of the j-th target label variable. The Monte Carlo method are used to estimate the expected value of $H(T)$ via sampling batch data and computing the batch entropy $\mathcal{L}_b$, which measures the uncertainty or diversity of the predicted probability distribution over the label space. The regularization term encourages uniformity across different classes in the latent space. This approach allows for a balanced learning process across different labels, preventing the model from excessively emphasizing certain prevalent features within the training data that might not accurately represent the true data distribution.

Structured Probabilistic Coding We incorporate the structured regularization from the target label space into the probabilistic coding framework, named Structured Probabilistic Coding (SPC). The total objective of SPC can be:

$$\begin{aligned} \mathcal{L}_{SPC} = \mathcal{L}_{PC} - \gamma H(T) &\gtrapprox \mathcal{L}_{PC} - \gamma\mathcal{L}_b \\ &= \mathbb{E}_{t\sim p_\theta(t|x)}[-\log q(y|t)] + \beta KL(p_\theta(t|x); r(t)) \\ &\quad + \gamma\sum_{j=1}^{|\mathcal{Y}|}\overline{p_{\cdot,j}}\log\left(\overline{p_{\cdot,j}}\right), \end{aligned} \tag{5}$$

where $\gamma > 0$ is a hyperparameter controlling the strength of the regularization. The first two terms combine probabilistic encoding and task prediction into one encoder module with the network θ. The last regularization term promotes class-level uniformity in the latent space under the multivariate Gaussian distribution. Totally, the goal of SPC is to maintain the Gaussian structure of the latent code, as well as achieve the best possible coverage of the hidden space with uniformity across classes.

Applications for Downstream Tasks We apply the SPC framework to enhance the generalization ability of pre-trained language models for various natural language understanding (NLU) tasks. Due to the ability of learning informative and compact representations, the proposed SPC framework is well-suited for classification and regression tasks. For classification tasks, the lower bound of $I(T;Y)$ can amount to a classic cross-entropy loss (Achille and Soatto 2018; Amjad and Geiger 2019). Similarly, for regression tasks, the lower bound of $I(T;Y)$ can be equivalent to a classic mean squared error loss.

Experiments

Experimental Setups

Datasets and Downstream Tasks We conduct experiments on various classification and regression tasks. Concretely, following Barbieri et al. (2020), we experiment on 7 classification tasks about tweet analysis on social media, i.e., **EmojiEval** (Barbieri et al. 2018), **EmotionEval** (Mohammad et al. 2018), **HatEval** (Basile et al. 2019), **IronyEval** (Hee, Lefever, and Hoste 2018), **OffensEval** (Zampieri et al. 2019), **SentiEval** (Rosenthal, Farra, and Nakov 2017), and **StanceEval** (Mohammad et al. 2016). To better evaluate the generalization of the method for cross-domain scenes, we also experiment on 3 emotion-related datasets from different domains, i.e., **ISEAR** (Scherer and Wallbott 1994), **MELD** (Poria et al. 2019), and **GoEmotions** (Demszky et al. 2020). Besides, we experiment on 2 regression benchmarks, i.e., **STS-B** (Cer et al. 2017) and **CLAIRE** (Roth, Anthonio, and Sauer 2022). See the appendix for more descriptions of datasets and tasks.

Comparison Methods We compare against the 4 universal models (i.e., SVM, FastText, BiLSTM, and GPT-3.5) and 7 representative deep representation learning technologies (i.e., CE/MSE, CE+CP, CE/MSE+AT, CE+SCL, VIB, MINE-IB, and MEIB). VIB, MINE-IB, and MEIB belong to probabilistic embedding methods, while the others typically belong to deterministic embedding methods. For these representation learning technologies, we use pre-trained language models, i.e., BERT (Devlin et al. 2019), and RoBERTa (Liu et al. 2019), as the backbone models for fine-tuning on downstream tasks. Concretely, we use *bert-base-uncased*[2] and *roberta-base*[2] to initialize BERT and RoBERTa for fine-tuning on downstream tasks, respectively.

SVM (Cortes and Vapnik 1995) is a machine learning algorithm with a hinge loss that aims to find the best hyperplane to separate data points into different classes. **FastText** (Joulin et al. 2017) is an efficient text classification method with negative log-likelihood loss based on n-gram features and a hierarchical softmax. **BiLSTM** is a bidirectional recurrent neural network (Hochreiter and Schmidhuber 1997) that can be used for classification with cross-entropy loss. **GPT-3.5**[3] is an enhanced generative pre-trained transformer model based on text-davinci-003, optimized for chatting.[4]

CE/MSE means a fine-tuned baseline with a cross-entropy (CE) loss for classification tasks or a mean squared error (MSE) loss for regression tasks. **CE+CP** (Pereyra et al. 2017) is an entropy regularization method that fits a deterministic network by optimizing an objective that combines the CE loss with a confidence penalty term. **CE/MSE+AT** (Miyato, Dai, and Goodfellow 2017) uses CE/MSE with classical adversarial training. **CE+SCL** (Gunel et al. 2021) combines CE and supervised contrastive learning (SCL) (Khosla et al. 2020). SCL allows for multiple positives per anchor, thus adapting contrastive learning to the fully supervised setting. **VIB** (Alemi et al. 2017; Mahabadi, Belinkov, and Henderson 2021) is an efficient variational estimation method of the information bottleneck (IB) principle (Tishby and Zaslavsky 2015). **MINE-IB** (Belghazi et al. 2018) is an neural estimation method of the IB principle with a continuous setting. **MEIB** (An, Jammalamadaka, and Chong 2023) is a variational approach to stochastic embedding in which maximum conditional entropy acts as the bottleneck. MEIB encourages obvious inputs that can be easily classified to take broader embedding areas by assigning larger entropy.

Evaluation Metrics We use the same evaluation metric from the original tasks. For the evaluation on classification tasks, the macro-averaged F1 over all classes is applied in most cases. There are three exceptions: stance (macro-averaged of F1 of favor and against classes), irony (F1 of ironic class), and sentiment analysis (macro-averaged recall). Following Barbieri et al. (2020), we report a global metric based on the average of all dataset-specific metrics. For the evaluation on regression tasks, we apply both Pearson and Spearman correlation coefficients. Besides, the t-test (Kim 2015) is used to verify the statistical significance of the differences between the results of our SPC and the best non-SPC method on the current dataset.

Implementation Details All experiments are conducted on a single NVIDIA Tesla A100 80GB card. The validation sets are used to tune hyperparameters and choose the optimal model. For each method, we run five random seeds and report the average result of the test sets. Besides, we conduct experiments using an epoch number of 20, a total batch size of 128, and a maximum token length of 128. The maximum patience for early stopping is set to 5 epochs. The network parameters are optimized by using Adamax optimizer (Kingma and Ba 2015) with the learning rate of $5e^{-5}$, the weight decay coefficient of $\{0, 0.01, 0.001\}$. For SPC, the trade-off parameter β and γ are searched from $\{0.001, 0.01, 0.1, 1, 10\}$ respectively. More experimental details are listed in the Appendix.

Overall Results

Performance on Classification Tasks The overall results for 10 classification tasks are summarized in Table 1. Our SPC consistently obtains the best average performance over comparison methods. When using BERT and RoBERTa backbones, SPC can enhance average performance by **+3.1%** and **+3.9%** compared to CE for all classification tasks, respectively. The results indicate the good generalization ability of our method to unseen test sets and show the superiority on classification tasks. We notice that SPC achieves big improvements for HatEval and IronyEval, i.e., **+14.3%** macro-F1 scores and **+7.3%** F1 scores of the ironic class, respectively. In HateEval, there is a topic distribution disparity between the validation set and the test set. Additionally, the IronyEval task requires complex semantic understanding, with subtle differences between ironic and non-ironic texts. These results indicate that our SPC has a good generalization capability on the above specific scenarios, i.e., topic shifts and subtle semantic labels.

Performance on Regression Tasks Table 3 presents the overall results of comparison methods in terms of Spearman and Pearson correlation coefficients for two regression tasks. SPC obtains better regression results on both datasets. Besides, when using RoBERTa backbone, compared to MSE, SPC achieves **+1.7%** absolute improvements in terms of the average performance. This demonstrates the superiority and generalization of SPC to unseen test sets on regression tasks.

Ablation Study

We conduct ablation studies by removing the structured regularization (w/o S.) and probabilistic coding (w/o P.). For classification, Table 2 shows the ablation results on all tasks.

[2]https://huggingface.co/

[3]https://chat.openai.com

[4]We present the zero-shot results of the GPT-3.5-turbo snapshot from June 13th 2023 based on specific inputs, including task descriptions, task instructions, and evaluation texts.

Methods	EmojiEval	EmotionEval	HatEval	IronyEval	OffensEval	SentiEval	StanceEval	ISEAR	MELD	GoEmotions	**Avg.**
SVM†	29.30	64.70	36.70	61.70	52.30	62.90	67.30	-	-	-	-
FastText†	25.80	65.20	50.60	63.10	73.40	62.90	65.40	-	-	-	-
BiLSTM†	24.70	66.00	52.60	62.80	71.70	58.30	59.40	-	-	-	-
GPT-3.5	6.34±0.01	73.23±0.18	48.30±0.11	**66.81**±3.26	63.71±0.13	40.40±3.13	39.45±0.10	67.22±0.09	41.46±0.11	25.21±0.08	47.21
BERT backbone											
CE	22.30±0.60	76.05±1.41	44.67±1.78	59.38±3.01	80.16±1.26	70.54±0.44	65.21±0.71	67.17±0.78	39.80±0.84	46.29±0.79	57.16
CE+CP	21.91±0.71	76.28±1.20	45.97±2.93	64.06±2.41	78.99±1.57	70.68±0.31	65.83±0.39	67.20±0.95	39.54±1.61	46.39±0.63	57.69
CE+AT	22.93±0.70	75.08±1.23	46.30±3.61	64.23±2.04	79.68±1.59	70.55±0.57	66.46±1.13	65.70±0.69	39.84±0.38	47.37±0.54	57.81
CE+SCL	21.72±0.51	75.43±1.37	45.86±1.15	65.39±2.46	80.20±0.56	70.70±0.79	65.34±0.60	67.54±0.64	40.00±1.96	46.50±0.46	57.87
VIB	21.31±0.62	77.37±0.71	45.99±1.93	63.82±1.00	80.37±1.11	70.39±0.31	65.43±0.60	67.24±0.57	38.52±0.51	45.89±1.10	57.63
MINE-IB	21.29±0.31	76.60±0.41	47.64±2.11	65.86±2.57	78.67±2.28	69.85±0.54	65.35±0.88	67.62±0.40	41.23±0.67	46.87±0.42	58.10
MEIB	21.87±0.73	76.70±0.82	48.27±1.72	65.87±2.14	80.49±0.81	70.55±0.57	65.59±1.58	67.44±0.50	39.30±0.61	46.26±0.81	58.23
SPC	24.19±1.55	77.15±0.73	57.48±2.99	65.85±1.07	80.65±0.78	70.74±0.12	67.17±1.08	68.94±0.35	42.68±0.94	47.62±1.38	60.25
RoBERTa backbone											
CE	30.25±1.32	77.41±1.33	45.49±4.70	57.99±4.96	78.74±2.20	71.80±0.93	66.78±1.34	70.00±0.45	39.23±0.41	46.64±1.15	58.43
CE+CP	31.12±0.84	77.54±0.70	48.59±3.28	58.75±6.19	79.50±0.98	72.82±0.29	66.89±1.67	70.58±0.71	40.74±0.89	47.98±0.65	59.45
CE+AT	32.00±0.93	77.30±1.07	44.71±4.76	60.17±3.17	79.81±1.11	72.51±0.44	67.81±0.95	70.97±0.68	40.10±0.60	47.89±1.21	59.33
CE+SCL	31.09±1.85	76.98±2.02	49.51±2.86	60.71±4.23	80.39±0.83	**73.16**±0.44	66.73±1.54	70.26±0.45	40.64±1.02	47.87±0.86	59.72
VIB	29.71±0.79	77.99±0.86	49.39±3.08	59.93±4.57	79.63±0.66	72.81±0.39	68.40±0.52	70.74±0.44	38.94±0.55	46.23±0.18	59.38
MINE-IB	31.70±0.45	78.79±0.58	46.39±2.82	57.39±8.27	79.76±0.67	72.85±0.56	67.27±1.00	70.15±0.58	41.80±2.14	48.88±1.04	59.50
MEIB	29.94±1.30	78.73±0.90	49.34±2.42	60.54±2.70	79.68±0.98	72.78±0.29	67.89±1.70	70.86±0.61	39.00±0.37	47.18±1.15	59.59
SPC	**32.54***±0.48	**79.01**±0.61	**59.80***±1.32	65.31±1.91	**80.98**±1.36	72.96±0.22	**69.02***±0.63	**71.01***±0.59	**43.99***±0.29	**48.92**±1.83	**62.35**

Table 1: Classification evaluation (%) on 10 benchmark datasets. † means the results are from Barbieri et al. (2020). For other methods, we run five random seeds and report the average result on test sets. BERT and RoBERTa are the backbone models for deep representation learning technologies. Best results for each dataset are highlighted in bold. * represents statistical significance over state-of-the-art scores under the t test ($p < 0.05$).

Methods	EmojiEval	EmotionEval	HatEval	IronyEval	OffensEval	SentiEval	StanceEval	ISEAR	MELD	GoEmotions	**Avg.**
SPC	**32.54**±0.48	**79.01**±0.61	**59.80**±1.32	**65.31**±1.91	**80.98**±1.36	**72.96**±0.22	**69.02**±0.63	**71.01**±0.59	**43.99**±0.29	**48.92**±1.83	**62.35**
- w/o S.	30.98±0.89	78.60±0.54	56.64±8.75	62.12±6.97	79.16±1.28	72.23±0.77	68.90±0.60	70.79±0.22	43.75±0.67	48.84±1.78	61.20
- w/o S. & P.	30.25±1.32	77.41±1.33	45.49±4.70	57.99±4.96	78.74±2.20	71.80±0.93	66.78±1.34	70.00±0.45	39.23±0.41	46.64±1.15	58.43

Table 2: Ablation results (%) on classification tasks. w/o S. indicates removing the structured regularization. w/o P. refers to removing probabilistic coding. We experiment with RoBERTa backbone.

Methods	STS-B		CLAIRE		Avg.
	Spearman	Pearson	Spearman	Pearson	
MSE	88.33±0.32	88.80±0.36	50.37±5.90	49.10±5.74	69.15
MSE+AT	88.40±0.50	89.01±0.37	53.09±0.64	51.87±0.65	70.59
VIB	88.45±0.50	89.01±0.40	52.86±0.88	51.66±0.78	70.49
MEIB	88.61±0.14	89.13±0.17	52.85±0.72	51.39±0.81	70.50
SPC	**88.71**±0.19	**89.31**±0.24	**53.11**±0.95	**52.21**±0.81	**70.84**

Table 3: Regression evaluation (%) on 2 benchmark datasets with RoBERTa backbone. For each method, we run five random seeds and report the average result on test sets.

When removing the structured regularization, the ablated model obtains inferior performance in terms of all classification metrics. When further removing probabilistic coding, the results decline significantly. It reveals the effectiveness of structured regularization and probabilistic coding. For regression, since the label space is a one-dimensional real-valued score, our SPC is degraded to probabilistic coding. The ablation removing probabilistic coding is equivalent to MSE. From Table 3, the average performance declines 1.7% on regression metrics, which confirms the effectiveness of probabilistic coding for regression.

Generalization Evaluation

We further evaluate the generalization capability of SPC under the following two settings: training with limited data and testing in out-of-distribution (OOD) scenarios.

Comparison under Different Training Size We experiment under different ratios of the training set to evaluate the generalization when training with limited data. Specifically, given a predefined ratio (e.g., 20%) and a random seed, we randomly sample from the original training set. We obtain 5 training subsets by independently and repeatedly sampling five times from the original training set with 5 different random seeds. All methods are trained on these 5 subsets of the training set, and we report the average results on the test set. Figure 2 shows results of CE, VIB, MEIB, and our SPC against different sizes of training set with RoBERTa backbone. compared to CE, VIB, and MEIB, SPC achieves superior performance on most datasets against different ratios of the training set. It indicates that SPC can enhance the generalization ability of pre-trained language models, even when dealing with limited training data.

Evaluation on Out-of-Distribution We choose emotion-related benchmarks including EmotionEval, ISEAR,

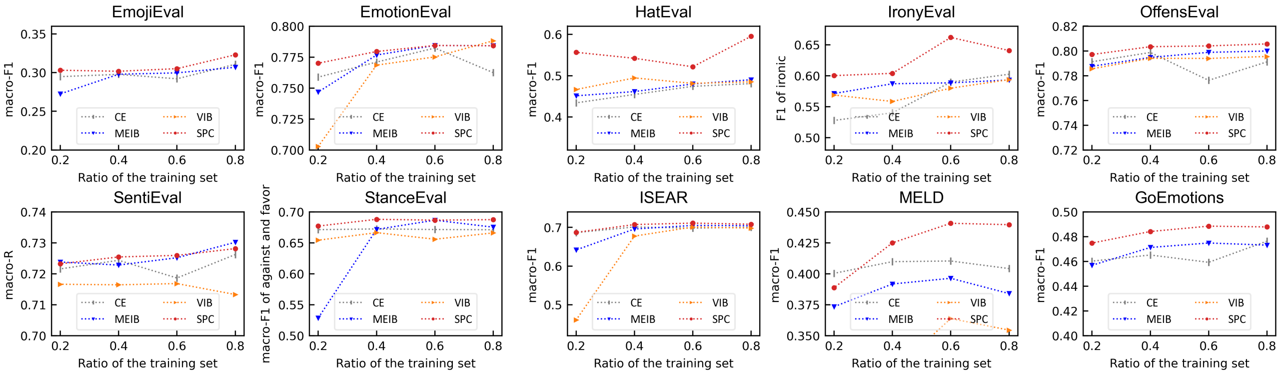

Figure 2: Results of different methods against different sizes of training set with RoBERTa backbone.

Methods	EmotionEval → GoEmotions	ISEAR → GoEmotions	MELD → GoEmotions	Avg.
CE	73.79±2.57	42.99±2.10	30.71±0.54	49.16
CE+AT	72.54±3.89	44.11±1.44	32.05±1.69	49.57
VIB	74.73±3.52	41.88±1.65	30.50±1.05	49.03
MEIB	75.55±2.05	42.10±0.61	30.11±1.33	49.25
SPC	**77.47**±2.46	**44.36**±1.29	**33.95**±1.16	**51.93**

Table 4: Out-of-distribution evaluation results (%). For instance, "EmotionEval → GoEmotions" refers to training the model on the training set of EmotionEval and making predictions using the test set of GoEmotions. We experiment with RoBERTa backbone. We run five random seeds and report the average results on test sets of target domains. Labels that do not appear in the training corpus are not evaluated.

MELD, and GoEmotions, which aim to predict the emotional state but are collected from different domains. To implement OOD scenarios, we train the model on the original training set from a source domain, select the best model based on the validation set of the source domain, and test on the test set of a target domain. To avoid the interference of label mapping bias between different taxonomies, each model is trained on the dataset with coarse-grained taxonomy to predict the label for another dataset with fine-grained taxonomy. Table 4 shows the performance under OOD scenarios. Our SPC obtains the best results on all OOD settings. The fact exhibits SPC's better generalization capabilities in handling OOD scenarios across different domain shifts. It indicates that SPC can better control and utilize randomness and uncertainty of data under the probabilistic coding framework. Moreover, the structure regularization from the target task space makes the learned probabilistic representations better reflect task information and generalize the model to new data.

Robustness Evaluation

We experiment to demonstrate the robustness by assessing how well the models can handle noisy labels. It is crucial for real-world scenarios where data can often be unreliable. Specifically, we randomly choose 10%, 20%, 30% of training data and flip their labels to any category randomly with equal probability. We run experiments five times and compute the mean and standard variance of the results. As shown in Table 5, under all settings, SPC consistently outperforms CE, VIB, and MEIB. It indicates that SPC performs more robustly on noisy training data. Besides, compared to CE, SPC improves average performance by **+1.9%**, **+2.2%**, and **+1.6%** with noise ratio of 10%, 20%, and 30% on classification tasks. The results prove that SPC can better control and utilize randomness and uncertainty of data.

Representation Quality Evaluation

To assess the quality of the representations, we evaluate the clustering performance of output representations obtained by different optimization objectives. Following Hu et al. (2023), we apply silhouette coefficient (SC) and adjusted rand index (ARI) to measure the clustering ability relevant to input data and target labels, respectively. Figure 3 shows SC and ARI of representations learned by various learning objectives. According to the results, SPC achieves higher ARI or SC values compared to other objectives (CE, VIB, and MEIB) across most datasets. It suggests that SPC effectively achieves a balance between data encoding and task prediction, thereby promoting the generalization of pre-trained language models for downstream tasks.

Related Work

Representation learning provides low-dimensional representations of the data, which can effectively capture the data features and serve for various tasks. According to the nature of embeddings, it can be broadly categorized into deterministic embedding and probabilistic embedding.

Deterministic Embedding Deterministic embedding maps each data point to a fixed vector. The representative works include entropy regularization (Pereyra et al. 2017), adversarial training (Miyato, Dai, and Goodfellow 2017), and contrastive learning (Gunel et al. 2021; Hu et al. 2023).

Probabilistic Embedding Probabilistic embedding (Vilnis and McCallum 2015) learns a probability distribution that maps each data point to a distribution. The probabilistic approach can better capture the complexity and uncertainty of data, handle redundant information, and provide better

Methods	Noisy	EmojiEval	EmotionEval	HatEval	IronyEval	OffensEval	SentiEval	StanceEval	ISEAR	MELD	GoEmotions	**Avg.**
CE	10%	30.66±0.89	78.15±0.88	47.06±5.40	56.90±4.58	79.46±0.80	72.36±0.74	67.39±1.86	70.40±0.97	42.01±1.94	47.85±1.08	59.22
VIB	10%	30.74±0.48	77.78±2.05	47.64±1.57	58.66±10.60	79.96±0.73	72.13±0.54	67.54±1.20	70.85±0.33	38.63±0.89	47.30±1.65	59.12
MEIB	10%	31.02±0.47	**78.94**±0.46	49.28±4.58	57.21±8.07	**80.19**±0.83	72.09±0.68	68.26±0.68	70.85±0.38	38.67±0.97	46.93±1.06	59.34
SPC	10%	**32.25**±0.69	78.88±0.47	**56.13**±5.36	**58.88**±4.94	80.14±0.28	**72.76**±0.06	**68.57**±1.01	**71.10**±0.62	**43.90**±1.13	**49.03**±1.41	**61.16**
CE	20%	31.96±0.88	77.01±1.51	49.12±0.72	60.82±3.56	79.54±1.64	72.06±0.63	68.49±1.20	70.32±0.26	40.16±1.94	47.78±0.84	59.73
VIB	20%	30.46±0.59	**79.00**±0.49	47.91±2.20	60.67±4.82	79.15±1.22	72.26±0.29	66.83±0.52	71.02±0.25	39.33±1.47	47.83±1.38	59.45
MEIB	20%	30.84±0.75	78.38±0.88	50.02±5.18	55.12±7.07	78.17±2.55	71.63±1.11	68.05±0.81	70.68±0.38	39.09±0.87	47.29±1.22	58.93
SPC	20%	**32.51**±0.83	77.97±1.12	**55.41**±6.00	**66.40**±4.26	**80.33**±0.48	**72.50**±0.55	**68.89**±1.60	**71.10**±0.39	**43.96**±0.50	**50.26**±0.79	**61.93**
CE	30%	31.82±0.75	77.61±0.90	50.69±2.80	58.90±11.45	78.11±2.07	70.15±0.5	69.07±1.07	70.74±0.56	40.61±2.06	47.76±2.29	59.55
VIB	30%	30.85±0.53	**78.23**±0.79	48.22±1.97	58.81±8.84	79.38±0.62	72.15±0.52	67.59±0.93	70.27±0.74	38.71±1.19	47.16±1.32	59.14
MEIB	30%	30.74±0.87	77.99±0.69	49.98±4.00	57.57±5.19	72.53±5.53	71.83±0.40	67.88±0.68	69.86±1.24	39.39±1.06	47.43±1.52	58.52
SPC	30%	**32.27**±0.48	78.13±1.13	**56.04**±7.44	**59.27**±8.56	**80.32**±0.53	**72.44**±0.36	**69.77**±0.93	**70.91**±0.30	**43.29**±0.53	**49.39**±0.64	**61.18**

Table 5: Results (%) against different ratios of label noises. RoBERTa is applied as the model backbone.

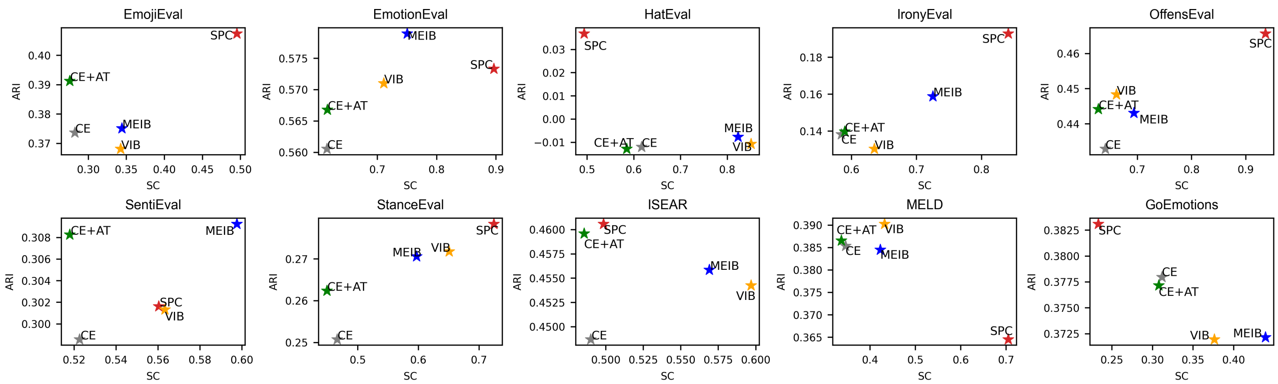

Figure 3: Clustering performances of the output representations learned by different optimization objectives. Silhouette coefficient (SC) and adjusted rand index (ARI) are used to measure data-related and task-related clustering abilities, respectively. We experiment with RoBERTa backbone.

discriminative representations. It has been applied to various domains such as computer vision (Oh et al. 2019; Shi and Jain 2019; Chang et al. 2020) and natural language processing (Mahabadi, Belinkov, and Henderson 2021; Wang et al. 2021; Hu et al. 2022).

Most probabilistic embedding methods (Kingma and Welling 2014; Alemi et al. 2017; Higgins et al. 2017; Fischer 2020; An, Jammalamadaka, and Chong 2023) are (or can be) built upon the principle of information bottleneck (IB) theory (Tishby, Pereira, and Bialek 1999; Tishby and Zaslavsky 2015). The principle aims to find a maximally compressed representation of the input that maximally preserves information about the output, striking a balance between compression and prediction. VIB (Alemi et al. 2017) is an efficient variational estimation method of the IB principle. For tractable application of IB in a continuous setting, Belghazi et al. (2018) propose a mutual information neural estimation method with IB principle, denoted as MINE-IB. And Ragonesi et al. (2021) employ MINE-IB to learn unbiased representations. Fischer (2020) and Ramé and Cord (2021) introduce a conditional mutual information term to alleviate the over- or under-compression issue of traditional IBs. Moreover, variational autoencoder (VAE) (Kingma and Welling 2014) is a special case of an unsupervised VIB and can be used to encourage disentangled representations (Higgins et al. 2017). Hu et al. (2022) apply VAE to the masked pre-training process for learning diverse and well-formed contextual representations. Recently, An, Jammalamadaka, and Chong (2023) use the conditional entropy of the stochastic embedding as a confidence indicator and encourage the model to assign larger variance to more certain inputs.

Conclusion

This paper proposes a new supervised representation learning framework SPC, an encoder-only probabilistic coding technology with a structured regularization from the target space. By extracting compact and informative representations from input related to the target task, SPC can enhance the generalization ability of pre-trained language models for better language understanding. Firstly, an encoder-only probabilistic coding technology simultaneously performs variational encoding and task prediction. Then, a structured regularization is introduced to control probability distribution and promote uniformity across classes in the latent space. Experiments on 12 benchmarks show that SPC achieves the best performance on various classification and regression tasks. The results demonstrate that SPC can enhance the generalization capability, robustness to label noise, and the clustering quality of output representations.

Appendix Overview

In this supplementary material, we provide: (i) a detailed description of experimental setups, and (ii) supplementary experiments.

Experimental Setups

Datasets and Downstream Tasks

We conduct extensive experiments on various natural language understanding tasks including 10 classification tasks, and 2 regression tasks, as shown in Table 6. The descriptions of each dataset and task are listed as follows:

Classification Tasks

- **EmojiEval** (Barbieri et al. 2018) is designed for emoji prediction, which aims to predict its most likely emoji given a tweet. Its label set comprises 20 different emoji.
- **EmotionEval** (Mohammad et al. 2018) involves detecting the emotion evoked by a tweet and is based on the Affects in Tweets conducted during SemEval-2018. Following Barbieri et al. (2020), the most common four emotions (i.e., anger, joy, sadness, and optimism) are selected as the label sets.
- **HatEval** (Basile et al. 2019) stems from SemEval-2019 Hateval challenge and is used to predict whether a tweet is hateful towards immigrants or women.
- **IronyEval** (Hee, Lefever, and Hoste 2018) is from SemEval-2018 Irony Detection and consists of identifying whether a tweet includes ironic intents or not.
- **OffensEval** (Zampieri et al. 2019) is from SemEval-2019 OffensEval and involves predicting if a tweet contains any form of offensive language.
- **SentiEval** (Rosenthal, Farra, and Nakov 2017) comes from SemEval 2017 and includes data from previous runs (2013, 2014, 2015, and 2016) of the same task. The goal is to determine if a tweet is positive, negative, or neutral.
- **StanceEval** (Mohammad et al. 2016) involves determining if the author of a piece of text has a favorable, neutral, or negative position towards a proposition or target.
- **ISEAR** (Scherer and Wallbott 1994) is from International Survey On Emotion Antecedents And Reactions project and contains reports on seven emotions each by close to 3000 respondents in 37 countries on all 5 continents. It aims to predict the emotion reaction. Due to the lack of a predefined split in the original dataset, we randomly split the dataset into train/valid/test set in a ratio of 4:1:5 based on the label distribution.
- **MELD** (Poria et al. 2019) contains multi-party conversation videos collected from Friends TV series, where two or more speakers are involved in a conversation. It is used to detect emotions in each utterance.[5]
- **GoEmotions** (Demszky et al. 2020) is a corpus of comments from Reddit, with human annotations to 27 emotion categories or neutral. It is used for fine-grained emotion detection. In this work, we remove all multi-label samples (nearly 16%) in the dataset to better evaluate the multi-class classification performance.

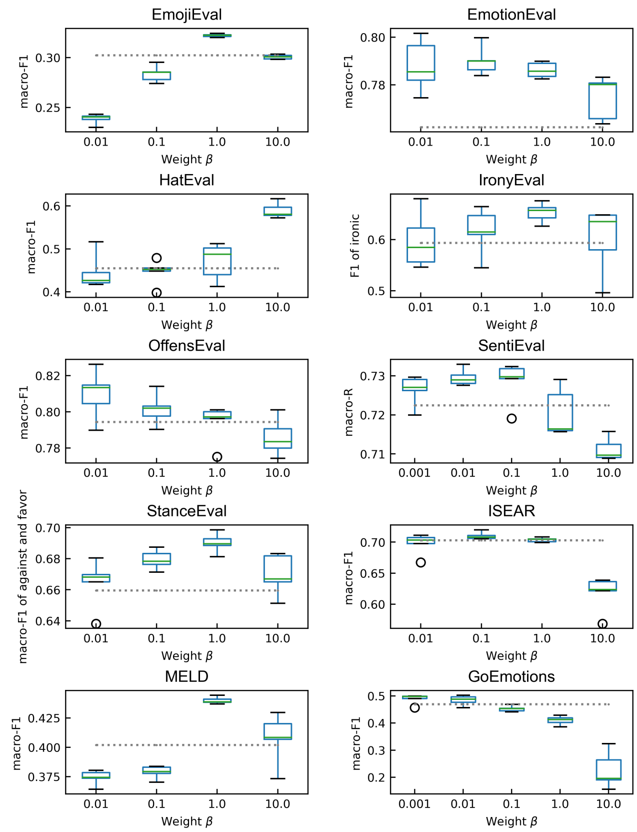

Figure 4: Performance against different trade-off weights β of probabilistic coding for classification tasks. The experiments are conducted with RoBERTa backbone. The grey line represents the results of CE baseline.

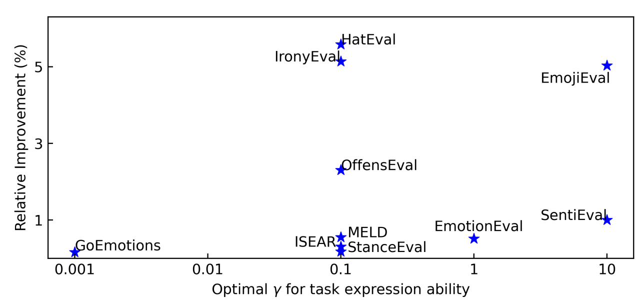

Figure 5: Performance of the optimal trade-off weight γ for classification tasks. We experiment with RoBERTa backbone. The Y-axis refers to relative improvements between SPC and its variant removing the structured regularization.

[5]The MELD dataset contains many types of context, including dialogue, speaker, and multi-modal data. Different from other task-oriented methods, e.g., DialogueCRN (Hu, Wei, and Huai 2021), this work only considers the context-free textual utterance to better evaluate sentence classification performance.

Dataset	Task	# Label	# Train	# Val	# Test	# Total
Classification						
EmojiEval	Emoji prediction	20	45,000	5,000	50,000	100,000
EmotionEval	Social emotion detection	4	3,257	374	1,421	5,052
HatEval	Hate speech detection	2	9,000	1,000	2,970	12,970
IronyEval	Irony detection	2	2,862	955	784	4,601
OffensEval	Offensive language detection	2	11,916	1,324	860	14,100
SentiEval	Sentiment analysis	3	45,389	2,000	11,906	59.295
StanceEval	Stance detection	3	2,620	294	1,249	4,163
ISEAR	Emotion reaction prediction	7	3,066	767	3,833	7,666
MELD	Conversational emotion recognition	7	9,989	1,109	2,610	13,708
GoEmotions	Fine-grained emotion detection	28	36,308	4,548	4,591	45,447
Regression						
STS-B	Semantic similarity prediction	-	7,000	1,500	1,400	9,900
CLAIRE	Plausible clarification ranking	-	19,975	2,500	2,500	24,975

Table 6: The statistics of all datasets.

Hyperparameter	EmojiEval	EmotionEval	HatEval	IronyEval	OffensEval	SentiEval	StanceEval	ISEAR	MELD	GoEmotions
Trade-off weight β	1	0.1	10	1	0.01	0.01	1	0.1	1	0.001
Trade-off weight γ	10	1	0.1	0.1	0.1	10	0.1	0.1	0.1	0.001
Weight decay	0	0	0.01	0	0	0	0.001	0.001	0	0
Dropout	0	0	0	0.2	0.2	0	0.2	0	0	0.2
Layer normalization	True	True	False	True	True	False	True	False	True	False

Table 7: Hyperparameters of the proposed SPC with RoBERTa backbone on classification tasks.

Hyperparameter	STS-B	CLAIRE
Trade-off weight β	0.01	0.1
Weight decay	0	0
Dropout	0	0
Layer normalization	False	False

Table 8: Hyperparameters of the proposed SPC with RoBERTa backbone on regression tasks.

Regression Tasks

- **STS-B** (Cer et al. 2017) is a collection of English sentence pairs drawn from news headlines, video and image captions, and natural language inference data. The semantic similarity prediction task is to predict the semantic textual similarity score from 0 (very dissimilar) to 5 (very similar) given each sentence pair.
- **CLAIRE** (Roth, Anthonio, and Sauer 2022) dataset consists of manually clarified how-to guides from wikiHow[6] with generated alternative clarifications and human plausibility judgements. The goal of plausible clarifications ranking task is to predict the continuous plausibility score on a scale from 1 (very implausible) to 5 (very plausible) given the clarification and its context. In our experiments, a special token pair (i.e., <e>and </e>) is introduced as the boundary of filler words.

Implementation Details

We report the detailed hyperparameter settings of SPC with RoBERTa backbone in Table 7 and Table 8. In the implementation of SPC, the hidden vector represents the output representation, and its dimension size is consistent with the dimension size of label space in each task.

For each baseline, we fine-tune the key parameters following the original paper for fair comparison and to obtain corresponding optimal performance. In addition, the temperature of GPT-3.5 is set to 0 for deterministic predictions.

[6]https://www.wikihow.com/

Supplementary Experiments

Parameter Analysis

In this part, we analyze what the trade-off weights β and γ control in our SPC.

Figure 4 shows results against different values of β. With the enhancement of the optimization strength of probabilistic coding ($\beta \uparrow$), SPC is prone to assign larger variance for noisy samples and small variance for high quality ones.

Figure 5 shows relative improvements between SPC and its ablated variant (i.e., w/o Structured) against the optimal γ. By introducing the underlying structured patterns related to the target task ($\gamma > 0$), SPC achieves varying degrees of relative improvements on all tasks, particularly in HatEval and IronyEval. A larger value of γ indicates that this type of task requires enhanced task-related learning ability.

Acknowledgements

This work was supported by the National Key Research and Development Program of China (No. 2022YFC3302102), and the National Funded Postdoctoral Researcher Program of China (No. GZC20232969). The authors thank the anonymous reviewers and the meta-reviewer for their helpful comments on the paper.

References

Achille, A.; and Soatto, S. 2018. Information dropout: Learning optimal representations through noisy computation. *IEEE TPAMI*, 40(12): 2897–2905.

Alemi, A. A.; Fischer, I.; Dillon, J. V.; and Murphy, K. 2017. Deep Variational Information Bottleneck. In *ICLR (Poster)*.

Amjad, R. A.; and Geiger, B. C. 2019. Learning representations for neural network-based classification using the information bottleneck principle. *IEEE TPAMI*, 42(9): 2225–2239.

An, S.; Jammalamadaka, N.; and Chong, E. 2023. Maximum Entropy Information Bottleneck for Uncertainty-aware Stochastic Embedding. In *CVPR Workshops*, 3809–3818.

Barbieri, F.; Camacho-Collados, J.; Anke, L. E.; and Neves, L. 2020. TweetEval: Unified Benchmark and Comparative Evaluation for Tweet Classification. In *EMNLP (Findings)*, 1644–1650.

Barbieri, F.; Camacho-Collados, J.; Ronzano, F.; Anke, L. E.; Ballesteros, M.; Basile, V.; Patti, V.; and Saggion, H. 2018. SemEval 2018 Task 2: Multilingual Emoji Prediction. In *SemEval@NAACL-HLT*, 24–33.

Basile, V.; Bosco, C.; Fersini, E.; Nozza, D.; Patti, V.; Pardo, F. M. R.; Rosso, P.; and Sanguinetti, M. 2019. SemEval-2019 Task 5: Multilingual Detection of Hate Speech Against Immigrants and Women in Twitter. In *SemEval@NAACL-HLT*, 54–63.

Belghazi, M. I.; Baratin, A.; Rajeswar, S.; Ozair, S.; Bengio, Y.; Hjelm, R. D.; and Courville, A. C. 2018. Mutual Information Neural Estimation. In *ICML*, volume 80, 530–539.

Cer, D. M.; Diab, M. T.; Agirre, E.; Lopez-Gazpio, I.; and Specia, L. 2017. SemEval-2017 Task 1: Semantic Textual Similarity Multilingual and Crosslingual Focused Evaluation. In *SemEval@ACL*, 1–14.

Chang, J.; Lan, Z.; Cheng, C.; and Wei, Y. 2020. Data uncertainty learning in face recognition. In *CVPR*, 5710–5719.

Cortes, C.; and Vapnik, V. 1995. Support-vector networks. *Machine learning*, 20: 273–297.

Demszky, D.; Movshovitz-Attias, D.; Ko, J.; Cowen, A. S.; Nemade, G.; and Ravi, S. 2020. GoEmotions: A Dataset of Fine-Grained Emotions. In *ACL*, 4040–4054.

Devlin, J.; Chang, M.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL-HLT*, 4171–4186.

Fischer, I. S. 2020. The Conditional Entropy Bottleneck. *Entropy*, 22(9): 999.

Goldfeld, Z.; and Polyanskiy, Y. 2020. The Information Bottleneck Problem and its Applications in Machine Learning. *IEEE J. Sel. Areas Inf. Theory*, 1(1): 19–38.

Gunel, B.; Du, J.; Conneau, A.; and Stoyanov, V. 2021. Supervised Contrastive Learning for Pre-trained Language Model Fine-tuning. In *ICLR*.

Hee, C. V.; Lefever, E.; and Hoste, V. 2018. SemEval-2018 Task 3: Irony Detection in English Tweets. In *SemEval@NAACL-HLT*, 39–50.

Higgins, I.; Matthey, L.; Pal, A.; Burgess, C. P.; Glorot, X.; Botvinick, M. M.; Mohamed, S.; and Lerchner, A. 2017. beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework. In *ICLR (Poster)*.

Hochreiter, S.; and Schmidhuber, J. 1997. Long short-term memory. *Neural computation*, 9(8): 1735–1780.

Hu, D.; Bao, Y.; Wei, L.; Zhou, W.; and Hu, S. 2023. Supervised Adversarial Contrastive Learning for Emotion Recognition in Conversations. In *ACL*, 10835–10852.

Hu, D.; Hou, X.; Du, X.; Zhou, M.; Jiang, L.; Mo, Y.; and Shi, X. 2022. VarMAE: Pre-training of Variational Masked Autoencoder for Domain-adaptive Language Understanding. In *EMNLP (Findings)*, 6276–6286.

Hu, D.; Wei, L.; and Huai, X. 2021. DialogueCRN: Contextual Reasoning Networks for Emotion Recognition in Conversations. In *ACL/IJCNLP (1)*, 7042–7052.

Joulin, A.; Grave, E.; Bojanowski, P.; and Mikolov, T. 2017. Bag of Tricks for Efficient Text Classification. In *EACL*, 427–431.

Khosla, P.; Teterwak, P.; Wang, C.; Sarna, A.; Tian, Y.; Isola, P.; Maschinot, A.; Liu, C.; and Krishnan, D. 2020. Supervised Contrastive Learning. In *NeurIPS*.

Kim, T. K. 2015. T test as a parametric statistic. *Korean journal of anesthesiology*, 68(6): 540–546.

Kingma, D. P.; and Ba, J. 2015. Adam: A Method for Stochastic Optimization. In *ICLR (Poster)*.

Kingma, D. P.; and Welling, M. 2014. Auto-Encoding Variational Bayes. In *ICLR*.

Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; and Stoyanov, V. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *CoRR*, abs/1907.11692.

Mahabadi, R. K.; Belinkov, Y.; and Henderson, J. 2021. Variational Information Bottleneck for Effective Low-Resource Fine-Tuning. In *ICLR*.

Miyato, T.; Dai, A. M.; and Goodfellow, I. J. 2017. Adversarial Training Methods for Semi-Supervised Text Classification. In *ICLR (Poster)*.

Mohammad, S. M.; Bravo-Marquez, F.; Salameh, M.; and Kiritchenko, S. 2018. SemEval-2018 Task 1: Affect in Tweets. In *SemEval@NAACL-HLT*, 1–17.

Mohammad, S. M.; Kiritchenko, S.; Sobhani, P.; Zhu, X.; and Cherry, C. 2016. SemEval-2016 Task 6: Detecting Stance in Tweets. In *SemEval@NAACL-HLT*, 31–41.

Oh, S. J.; Murphy, K. P.; Pan, J.; Roth, J.; Schroff, F.; and Gallagher, A. C. 2019. Modeling Uncertainty with Hedged Instance Embeddings. In *ICLR (Poster)*.

Pereyra, G.; Tucker, G.; Chorowski, J.; Kaiser, L.; and Hinton, G. E. 2017. Regularizing Neural Networks by Penalizing Confident Output Distributions. In *ICLR (Workshop)*.

Poria, S.; Hazarika, D.; Majumder, N.; Naik, G.; Cambria, E.; and Mihalcea, R. 2019. MELD: A Multimodal Multi-Party Dataset for Emotion Recognition in Conversations. In *ACL*, 527–536.

Ragonesi, R.; Volpi, R.; Cavazza, J.; and Murino, V. 2021. Learning Unbiased Representations via Mutual Information Backpropagation. In *CVPR Workshops*, 2729–2738.

Ramé, A.; and Cord, M. 2021. DICE: Diversity in Deep Ensembles via Conditional Redundancy Adversarial Estimation. In *ICLR*.

Rosenthal, S.; Farra, N.; and Nakov, P. 2017. SemEval-2017 Task 4: Sentiment Analysis in Twitter. In *SemEval@ACL*, 502–518. Association for Computational Linguistics.

Roth, M.; Anthonio, T.; and Sauer, A. 2022. SemEval-2022 Task 7: Identifying Plausible Clarifications of Implicit and Underspecified Phrases in Instructional Texts. In *SemEval@NAACL*, 1039–1049.

Scherer, K. R.; and Wallbott, H. G. 1994. Evidence for universality and cultural variation of differential emotion response patterning. *Journal of personality and social psychology*, 66(2): 310.

Shi, Y.; and Jain, A. K. 2019. Probabilistic Face Embeddings. In *ICCV*, 6901–6910.

Tishby, N.; Pereira, F. C.; and Bialek, W. 1999. The information bottleneck method. *arXiv preprint physics/0004057*.

Tishby, N.; and Zaslavsky, N. 2015. Deep learning and the information bottleneck principle. In *IEEE Information Theory Workshop*, 1–5.

Vilnis, L.; and McCallum, A. 2015. Word Representations via Gaussian Embedding. In *ICLR*.

Wang, B.; Wang, S.; Cheng, Y.; Gan, Z.; Jia, R.; Li, B.; and Liu, J. 2021. InfoBERT: Improving Robustness of Language Models from An Information Theoretic Perspective. In *ICLR*.

Zampieri, M.; Malmasi, S.; Nakov, P.; Rosenthal, S.; Farra, N.; and Kumar, R. 2019. SemEval-2019 Task 6: Identifying and Categorizing Offensive Language in Social Media (OffensEval). In *SemEval@NAACL-HLT*, 75–86.