Description-Enhanced Label Embedding Contrastive Learning for Text Classification

Kun Zhang, *Member, IEEE*, Le Wu, *Member, IEEE*, Guangyi Lv, Enhong Chen, *Senior Member, IEEE*, Shulan Ruan, Jing Liu, *Member, IEEE*, Zhiqiang Zhang, Jun Zhou, and Meng Wang, *Fellow, IEEE*

***Abstract*— Text classification is one of the fundamental tasks in natural language processing, which requires an agent to determine the most appropriate category for input sentences. Recently, deep neural networks have achieved impressive performance in this area, especially pretrained language models (PLMs). Usually, these methods concentrate on input sentences and corresponding semantic embedding generation. However, for another essential component: labels, most existing works either treat them as meaningless one-hot vectors or use vanilla embedding methods to learn label representations along with model training, underestimating the semantic information and guidance that these labels reveal. To alleviate this problem and better exploit label information, in this article, we employ self-supervised learning (SSL) in model learning process and design a novel self-supervised relation of relation (R^2) classification task for label utilization from a one-hot manner perspective. Then, we propose a novel relation of relation learning network (R^2-Net) for text classification, in which text classification and R^2 classification are treated as optimization targets. Meanwhile, triplet loss is employed to enhance the analysis of differences and connections among labels. Moreover, considering that one-hot usage is still short of exploiting label information, we incorporate external knowledge from WordNet to obtain multiaspect descriptions for label semantic learning and extend R^2-Net to a novel description-enhanced label embedding network (DELE) from a label embedding perspective. One step further, since these fine-grained descriptions may introduce unexpected noise, we develop a mutual interaction module to select appropriate parts from input sentences and labels simultaneously based on contrastive learning (CL) for noise mitigation. Extensive experiments on different text classification tasks reveal that R^2-Net can effectively improve the classification performance and DELE can make better use of label information and further improve the performance. As a byproduct, we have released the codes to facilitate other research.**

***Index Terms*— Contrastive learning (CL), label embedding, representation learning, text classification.**

Manuscript received 24 July 2022; revised 30 November 2022, 3 February 2023, and 19 April 2023; accepted 25 May 2023. This work was supported in part by the Young Scientists Fund of the National Natural Science Foundation of China under Grant 62006066; in part by the National Natural Science Foundation of China under Grant 61727809, Grant 61922073, and Grant 72188101; in part by the joint Funds of the National Natural Science Foundation of China under Grant U22A2094; and in part by the Fundamental Research Funds for the Central Universities under Grant JZ2021HGTB0075. *(Corresponding author: Le Wu.)*

Kun Zhang, Le Wu, and Meng Wang are with the School of Computer and Information, Hefei University of Technology, Hefei, Anhui 230029, China (e-mail: zhang1028kun@gmail.com; lewu.ustc@gmail.com; eric.mengwang@gmail.com).

Guangyi Lv is with the AI Lab, Lenovo Research, Beijing 100094, China (e-mail: lvgy1@lenovo.com).

Enhong Chen and Shulan Ruan are with the School of Computer Science and Technology, University of Science and Technology of China, Hefei 230026, China (e-mail: cheneh@ustc.edu.cn; slruan@mail.ustc.edu.cn).

Jing Liu is with the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China (e-mail: jliu@nlpr.ia.ac.cn).

Zhiqiang Zhang and Jun Zhou are with Ant Group Company Ltd., Hangzhou 310007, China (e-mail: lingyao.zzq@antfin.com; jun.zhoujun@antfin.com).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TNNLS.2023.3282020.

Digital Object Identifier 10.1109/TNNLS.2023.3282020

I. Introduction

AS ONE of fundamental tasks in natural language processing (NLP), text classification focuses on identifying the most appropriate category for sentences or the most suitable relation for sentence pairs. For example, paraphrase identification (PI) aims at identifying whether the sentence pair expresses the same meaning (yes or no) [1]. Natural language inference (NLI) targets at classifying input sentence pair into one of three relations (i.e., entailment, contradiction, and neutral) [2]. Question answering (QA) topic classification requires an agent to select the most suitable topic for a given question–answer pair [3]. Fig. 1(a) shows some examples with different relations from different tasks.

As a vital technology, text classification has been applied successfully to various NLP fields, e.g., sentiment analysis [4], [5], [6], information retrieval [7], QA [8], and dialog system [9]. Based on label usage, most of the existing works can be grouped into two categories. The first category is one-hot encoding of labels. Researchers usually focus on designing a deep model to learn representations for input text [10], [11], [12]. Then, a simple classifier is employed to predict the label distribution. A cross-entropy loss between the prediction and one-hot label encoding is finally adopted for model training [13], [14]. However, labels can reveal some common characteristics of examples within the same category [15] and provide rich semantic information as well as guidance for sentence semantics learning [16]. Treating labels as independent and meaningless one-hot vectors will cause potential information loss and have weaknesses in dealing with fine-grained interactions between text and labels. Gururangan et al. [17] observed that different relations among sentence pairs imply specific semantic expressions. Taking case 1 in Fig. 1(a) as an example, when constructing sentence pairs that are semantically contradictory, negation words (e.g., replacing "A lady" with "Nobody" in pair a and b2) are usually used. Moreover, replacing exact numbers with approximates

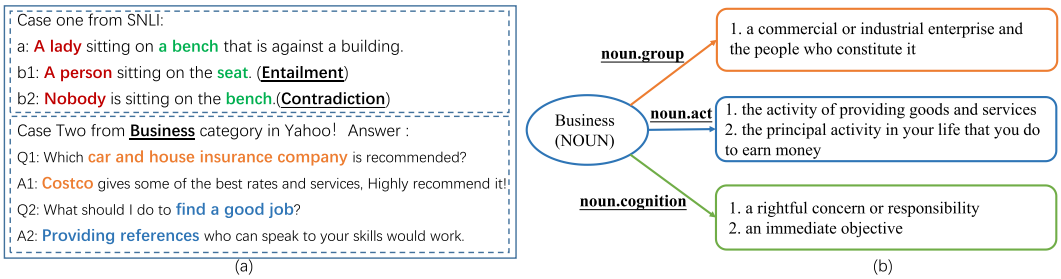


Fig. 1. (a) Some text classification examples: label signals can imply specific some semantic expressions of input sentences (e.g., in entailment pair (a and b1): "A lady" is replaced with "A person" and in contradiction pair (a and b2): "A lady" is replaced with "Nobody"). (b) Some fine-grained noun descriptions for label word "Business" from WordNet, which give detailed explanations about different attributes of label "Business."

can always generate "entailment" semantic relation. Therefore, the connection and differences among different relations (e.g., pairwise relation comparison) will be helpful to capture more implicit common semantic features, which is a promising direction to fully exploit label information for text classification. Then, the problem becomes how to design this type of signal and integrate it into the model learning process, which is one of the main focuses of this article.

The second category is label embedding method. Different from one-hot encoding methods, this type of work usually uses dense vectors to represent labels in the same semantic space as text embeddings. Then, attention mechanism is employed to measure the impact of label semantics on input text for classification. For example, Xiao et al. [18] proposed a label-specific attention network to enhance text representation learning. Zhang et al. [19] proposed to retrieve keywords from documents as pseudo descriptions for label embedding learning. Despite the inspiring performance, there still are some shortcomings. Specifically, vanilla embedding methods treat labels as general words and generate representations during model training. Knowledge-enhanced methods usually use a single description to enrich the embedding. They either only focus on the literal meanings of labels or describe labels in a coarse-grained manner, which is insufficient for precise and relevant label embedding learning. As shown in Fig. 1(b), even if a particular meaning of a word is utilized as label semantics, this meaning still has different attributes and can be described from different perspectives (e.g., noun.group and noun.act). Moreover, even if different input texts have the same labels, their semantics may associate with different aspects of the label. For example, for label "Business," noun.group attribute is used to identify QA(1), and noun.act attribute is used for QA(2) in case 2 in Fig. 1(a). Thus, it is promising to consider fine-grained knowledge (e.g., multiaspect descriptions from WordNet) to enhance label embedding. Meanwhile, different from key words extraction and single description usage, multiaspect descriptions may not all relate to the specific label meaning, which will import unanticipated noise and harm the precise label semantic representation. For example, noun.cognition attribute in Fig. 1(b) is not the option for the label "Business" used in Fig. 1(a), which will become noise and confuse models to understand the semantics of the label "Business." Therefore, another main contribution of this article is how to introduce fine-grained information (i.e., multiaspect descriptions) for better label embedding learning, as well as eliminating unexpected noise.

In order to mine the implicit information inside labels in a one-hot encoding manner, in our preliminary work [20], we propose a novel relation of relation learning network (R^2-Net) to fully exploit labels in a simple but effective way. We first utilize pretrained language model (PLM) encoder (e.g., BERT [21]) and convolutional neural network (CNN)-based encoder to model global and partial semantic meanings of input words and sentences separately. Then, inspired by self-supervised learning (SSL), we propose a self-supervised relation of relation (R^2) classification task to enhance the learning ability of R^2-Netso that interclass relations of input texts will be modeled comprehensively (i.e., differences among different label relations). Moreover, a triplet loss is used to measure the connections among the same classes (e.g., those inputs that have the same label will be represented much closer and vice versa further apart).

However, R^2-Net still mines label information in a one-hot encoding manner, which inhibits the potential of label utilization. To this end, in this article, we focus on label embedding and extend R^2-Net to a novel description-enhanced label embedding network (DELE), which incorporates fine-grained descriptions from WordNet as prior knowledge to boost the label embedding learning. Specifically, we first extract multiple descriptions from WordNet for each label word to enrich the label embedding learning. After that, considering that multiaspect descriptions may not all relate to the specific meaning of labels, we design a novel mutual interaction module based on contrastive learning (CL) framework to explore bidirectional interactions between input sentences and labels. Along this line, sentence representations can be leveraged to denoise multiaspect descriptions of labels and select the most relevant parts for better semantic representations. Meanwhile, label semantics can also be used to guide the importance selection of input sentences as existing methods do. Extensive experiments over different types of text classification tasks also prove that DELE can make better use of labels and do better classification.

In summary, the main contributions of this article lie in the following parts: 1) we develop a novel self-supervised R^2 task to mine the implicit information inside labels from a one-hot encoding perspective (preliminary work); 2) we propose to leverage multiaspect descriptions from WordNet to fully

exploit label information from label embedding perspective (extended work); 3) we also design a novel mutual interaction module based on CL for better description denoising and integrating (extended work); and 4) extensive experiments over different types of text classification tasks have been done to demonstrate the effectiveness of our proposed methods.[1]

The remainder of this article is organized as follows. Section II summarizes related work. Section III gives formal definitions of text classification and our proposed R^2 classification. Sections IV and V report technical details of our proposed R^2-Net and DELE. The experiments and detailed analysis are reported in Section VI. Finally, we discuss and conclude our work in Sections VII and VIII.

II. Related Work

In this section, we will introduce the related work, which is grouped into two lines of literature: 1) text classification, focusing on sentence semantic modeling and relation identification with different label usages, and 2) CL, introducing the recent progress of CL on text classification.

A. Text Classification

With the development of various neural networks such as CNN [22], GRU [23], and attention mechanism [5], [24], plenty of methods have been exploited for text classification on large datasets, such as SNLI [25], Quora [26], and SST-5 [27]. Usually, researchers employ neural networks [21], [28] to generate text representations. Then, a simple classifier, such as multilayer perceptron (MLP), is used to predict the label distribution. Next, a cross-entropy loss is applied for model training [13], [29]. Based on label usage, there are two categories: one-hot encoding methods and label embedding methods, which are summarized as follows.

For one-hot encoding methods, researchers focus on input text and treat labels as one-hot meaningless training signals. For example, Zeng et al. [30] focused on the input sentences and generated representations by extracting lexical- and sentence-level features. Zhang et al. [10] developed a DRr-Net to select important parts in a sentence precisely for text representation and classification. Similarly, Liu et al. [31] leveraged a sequential decision process with reinforcement learning to exploit the potential of sentences for classification. After PLMs (e.g., BERT [21] and RoBERTa [28]) have been proposed, this kind of learning paradigm has become much simpler and more effective. They have all achieved impressive performances. However, in most scenarios, treating labels as independent and meaningless one-hot training signals will ignore the implicit semantics and common feature indication of label words. This will cause information loss and limit the learning capability of representation methods.

To better exploit label information, label embedding methods are proposed. In practice, researchers treat the text classification problem as a label–text joint embedding problem, where labels are embedded in the same space as texts. Then, the attention mechanism is employed to measure semantic relations between labels and texts for classification [18], [32], [33], [34], [35]. Traditionally, vanilla embedding is the most common choice. For example, Du et al. [36] used vanilla embedding to represent label semantics from scratch and designed an EXAM to explicitly calculate matching scores between text and labels at the word level. To generate better label embedding, side information and relation modeling are taken into consideration. For side information, Zhu et al. [35] used term frequency-inverse document frequency (TF-IDF) to select the most effective words from documents as the descriptions of labels and leveraged the attention mechanism to fuse the input sentences and label information for better classification. Rivas Rojas et al. [37] extracted textual definitions from Oxford Dictionaries for label modeling. For relations, Guo et al. [14] developed a label confusion model to estimate the label dependency for better label exploration. In addition, there still exist other methods that leveraged label embedding to tackle multilabel classification [15], [38], hierarchical text classification [39], and so on.

Compared with existing work, our proposed work has the following main improvements. First, we propose to use multiaspect real sentences as descriptions for the fine-grained side information of labels. Second, we argued that additional descriptions would import unexpected noise and harm the model performance. Then, we developed a novel mutual interaction based on CL to achieve the fusing of input sentences and labels as well as alleviate the unexpected noise introduced by multiaspect descriptions.

B. Contrastive Learning

As a core component of SSL, CL has led to the state-of-the-art performance in the unsupervised training of deep learning models and achieved impressive performance in computer vision [40] and NLP [20], [41]. Traditionally, CL uses data augmentations to generate "positive" pairs and randomly selects "negative" samples from mini-batch without the consideration of labels. The target is to pull together an anchor and a positive sample in embedding space and push apart the anchor from many negative samples [42]. This paradigm can be treated as contrastive instance discrimination [43]. SimCLR [44], MoCo [45], and MAE [46] are representative works.

However, instance-level CL has some weaknesses in mining common features among instances with the same category. Therefore, researchers have proposed to consider side information for better sampling, such as supervised CL [42], [47] and cluster-based CL [48]. The most relevant work is supervised CL with the consideration of labels. For example, Khosla et al. [42] used labels to select positive examples with the same labels as the anchor example. Therefore, models can obtain multiple positive examples not only from data augmentation but also from the same category. One step further, Gao et al. [47] leveraged "Entailment" and "Contradiction" labels to select positive examples and negative examples simultaneously. Besides, Yang et al. [49], [50] developed a noise-robust CL loss for handling the false negative situations in CL for performance improvement. In our preliminary work [20], we designed an R^2 task to force the model to measure the label relations with CL. They have all made some

[1]https://wordnet.princeton.edu/

progress in improving CL performance and achieved promising results in downstream tasks. Nevertheless, these supervised CL-based methods still treat labels as one-hot signals and use these signals to directly select hard examples, underestimating the potential semantic information of labels, that is to say, there still is plenty of space for further improving text representation and classification with better label utilization methods.

III. Problem Statements

In this section, we will introduce the definition of text classification task, our proposed R^2 classification task, and necessary notations.

A. Text Classification

Given a word sequence denoted by $\boldsymbol{X} = \{x_1, x_2, \ldots, x_n\}$, where n is the length of the word sequence, as well as the label set $\mathcal{Y}(|\mathcal{Y}| = m)$, where the jth label is represented with one-hot representation $\boldsymbol{y}_j$ and m is the number of labels, the target is to learn a classifier ξ, which is capable of computing the conditional probability $P(\boldsymbol{y}|\boldsymbol{X}, \mathcal{Y})$ and predicting the most appropriate label for the input text

$$\begin{aligned} P(\boldsymbol{y}|\boldsymbol{X}, \mathcal{Y}) &= \xi(\boldsymbol{X}, \mathcal{Y}) \\ y^* &= \text{argmax}_{y \in \mathcal{Y}} P(\boldsymbol{y}|\boldsymbol{X}, \mathcal{Y}) \end{aligned} \tag{1}$$

where true label $y \in \mathcal{Y}$ indicates the semantic category of input text. For example, $\mathcal{Y} = \{\text{Yes}, \text{No}\}$ for PI task, and $\mathcal{Y} = \{\text{entailment, contradiction, neutral}\}$ for NLI task.

B. R^2 Classification

Previous study [17] has demonstrated that categories can be helpful to reveal some implicit patterns for text classification. Therefore, we propose a novel R^2 classification task to guide models to exploit category information precisely. Given two input texts $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$, the goal is to learn a classifying function $\mathcal{F}$ to identify whether these two input texts have the same semantic relation

$$\mathcal{F}(\boldsymbol{X}_1, \boldsymbol{X}_2) = \begin{cases} 1, & \text{if } \boldsymbol{y}_1 = \boldsymbol{y}_2 \\ 0, & \text{if } \boldsymbol{y}_1 \neq \boldsymbol{y}_2 \end{cases} \tag{2}$$

where $\boldsymbol{y}_1$ and $\boldsymbol{y}_2$ stand for one-hot representations of labels of two input texts.

Next, we will introduce the technical details of our proposed R^2-Net and DELE. Some necessary notations are listed in Table I for better illustration.

TABLE I
NOTATIONS USED IN R^2-NET AND DELE

Notation	Explanation
$\boldsymbol{X}$	The input text sequence
$\mathcal{Y}$	The label set
$\boldsymbol{H}$	Word embedding matrix for entire input text
$\boldsymbol{v}_g, \boldsymbol{v}_l$	Global and local representations for input text
$\boldsymbol{v}_*$	Different vector representations from different modules
$\alpha_1, \alpha_2, ..., \alpha_L$	Trainable weights for different layers in PLMs
$S^{(i)}$	Fine-grained description set for the i^{th} label
$\boldsymbol{E}_f$	Vanilla label embedding matrix
$\boldsymbol{E}$	Final label embedding matrix
$\boldsymbol{W}_*, \boldsymbol{U}_*, \boldsymbol{\omega}_*, \boldsymbol{b}_*$	Trainable parameters of our proposed models
$\boldsymbol{\beta}^l, \boldsymbol{\beta}^t$	Attention weights used in *DELE*

IV. Relation of Relation Learning Network

Fig. 2(a) reports the overall architecture of R^2-Net. To better describe the technical details of R^2-Net, similar to Section III, we elaborate them from two aspects: 1) text classification part and 2) R^2 classification part.

A. Text Classification Part

This part focuses on identifying the most suitable label for a given text. Specifically, we first utilize powerful PLMs [21], [28], such as BERT, to generate sentence semantic representation globally. Meanwhile, we develop a CNN-based encoder to capture keywords and phrase information from a local perspective. Thus, the input text sequence can be encoded in a comprehensive manner. Then, we leverage an MLP to predict the corresponding label.

1) Global Encoding: With the full usage of large corpus and multilayer transformers, PLMs (e.g., BERT [21]) have accomplished much progress in many NLP tasks. Thus, we take pretrained BERT as an example to describe how to generate sentence semantic representations for the input. Moreover, inspired by ELMo [51], we also use the weighted sum of all the hidden states of words from different transformer layers as the final contextual representations of input words in sentences.

Specifically, we first split input text into BPE tokens [52] and add special token "[CLS]" at the beginning and the end of word sequence. "[SEP]" token is also considered when there is more than one sentence. Then, BERT is utilized to generate word representation $\boldsymbol{H}$ and sentence representations $\boldsymbol{v}_g$. As shown in Fig. 2(b), suppose that there are L layers in BERT. Contextual word representations for input text sequence $\boldsymbol{X}$ are then a prelayer weighted sum of transformer block output, with weights $\alpha_1, \alpha_2, \ldots, \alpha_L$

$$\begin{aligned} \boldsymbol{h}_0^l, \boldsymbol{H}^l &= \text{BERT}_l(\boldsymbol{X}) \\ \boldsymbol{H} &= \sum_{l=1}^{L} \alpha_l \boldsymbol{H}^l, \quad \boldsymbol{v}_g = \boldsymbol{h}_0^L \end{aligned} \tag{3}$$

where $\boldsymbol{h}_0^l$ denotes the representation of first token "[CLS]" at the lth layer, $\boldsymbol{v}_g$ denotes the global semantic representation of the input, $\boldsymbol{H}$ represents the sequence features of the whole input, and α_l is the weight of the lth layer in BERT and will be learned during model training.

2) Local Encoding: The semantic relation within the text sequence is not only connected with the important words but also affected by the local information (e.g., phrase and local structure). Though BERT leverages multilayer transformers to perceive important words in the sentence pair, it still has some weaknesses in modeling local information. To alleviate these shortcomings, we develop a CNN-based local encoder to extract local information from the input.

Fig. 2(c) reports the local encoding structure. The input of this module is $\boldsymbol{H}$ from global encoding. We use convolution operations with different composite kernels (e.g., bi-gram and tri-gram) to process these features. Each operation with different kernels can model local patterns with different sizes.

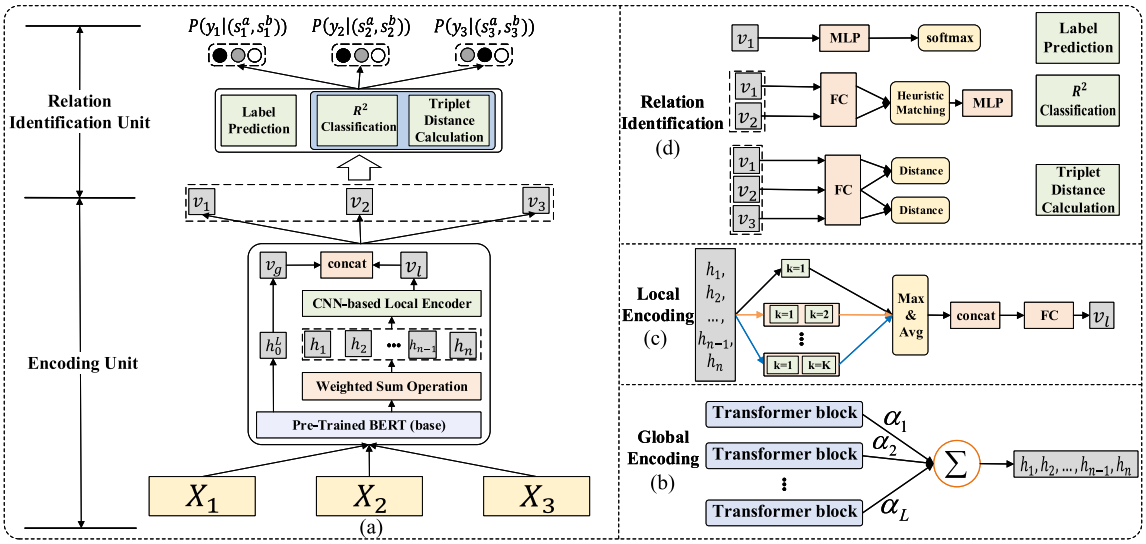


Fig. 2. (a) Overall structure of R^2-Net. (b) Using PLMs to obtain the representations of input words and sentences. (c) Our proposed CNN-based local encoding for the model ability enhancement on sentence partial information extraction. (d) Relation identification module for our proposed R^2 classification.

Next, we leverage average pooling and max pooling to enhance these local features and concatenate them before sending them to a nonlinear transformation. Suppose that we have K different kernel sizes. This process can be formulated as follows:

$$\begin{aligned} \boldsymbol{H}^k &= \mathrm{CNN}_k(\boldsymbol{H}), \quad k = 1, 2, \ldots, K \\ \boldsymbol{h}^k_{\max} &= \max\left(\boldsymbol{H}^k\right), \quad \boldsymbol{h}^k_{\mathrm{avg}} = \mathrm{avg}\left(\boldsymbol{H}^k\right) \\ \boldsymbol{h}_{\mathrm{concat}} &= \left[\boldsymbol{h}^1_{\max}; \boldsymbol{h}^1_{\mathrm{avg}}; \ldots; \boldsymbol{h}^K_{\max}; \boldsymbol{h}^K_{\mathrm{avg}}\right] \\ \boldsymbol{v}_l &= \mathrm{ReLu}(\boldsymbol{W}\boldsymbol{h}_{\mathrm{concat}} + \boldsymbol{b}) \end{aligned} \tag{4}$$

where CNN_k denotes the convolution operation with the kth kernel, $[\cdot;\cdot]$ is the concatenation operation, $\boldsymbol{v}_l$ represents the local semantic representation of the input, $\{\boldsymbol{W} \in \mathcal{R}^{d_p*(2K*d_p)}, \boldsymbol{b} \in \mathcal{R}^{d_p}\}$ are trainable parameters, d_p is the output size of pretrained model, and $\mathrm{ReLu}(\cdot)$ is the activation function.

After getting the global representation $\boldsymbol{v}_g$ and local representation $\boldsymbol{v}_l$, we investigate different fusion methods to integrate them, including simple concatenation, weighted concatenation, and weighted sum. Finally, we obtain that simple concatenation is flexible and can achieve comparable performance without extra training parameters. Thus, we employ concatenation $\boldsymbol{v} = [\boldsymbol{v}_g; \boldsymbol{v}_l]$ as the final semantic representation of input text sequence.

3) Label Prediction: This component is adopted to predict the label of input text, which is an essential text classification part. To be specific, the input of this component is semantic representation $\boldsymbol{v}$. We leverage a two-layer MLP to make the final classification, which can be formulated as follows:

$$P(y|\boldsymbol{X}) = \mathrm{MLP}_1(\boldsymbol{v}). \tag{5}$$

B. R^2 Learning Part

This part aims to properly and fully use comparison and CL of label information and help to improve the model performance. To achieve this goal, we employ two critical modules to analyze pairwise relation and triplet-based relation simultaneously.

1) R^2 Classification: Inspired by SSL methods in PLMs (e.g., MLM and NSP in BERT), we intend R^2-Net to make full use of label information in a similar way. Therefore, we develop a novel self-supervised R^2 classification task. Instead of just identifying the most suitable label, we plan to obtain more information about the input text sequence by analyzing the pairwise relation between semantic representations (i.e., $\boldsymbol{v}_1$ for $\boldsymbol{X}_1$ and $\boldsymbol{v}_2$ for $\boldsymbol{X}_2$). Since a learnable nonlinear transformation between representations and loss substantially improves the model performance [44], we first transfer $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$ with a nonlinear transformation. Then, we leverage heuristic matching [53], [54] to model the similarity and difference between $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$. Next, we send the matching result $\boldsymbol{u}$ to an MLP with one hidden layer for final classification. This process is formulated as follows:

$$\begin{aligned} \bar{\boldsymbol{v}}_1 &= \mathrm{MLP}_1(\boldsymbol{v}_1), \quad \bar{\boldsymbol{v}}_2 = \mathrm{MLP}_1(\boldsymbol{v}_2) \\ \boldsymbol{u} &= \left[\bar{\boldsymbol{v}}_1; \bar{\boldsymbol{v}}_2; \left(\bar{\boldsymbol{v}}_1 \odot \bar{\boldsymbol{v}}_2\right); \left(\bar{\boldsymbol{v}}_1 - \bar{\boldsymbol{v}}_2\right)\right] \\ P\left(\hat{y}|\boldsymbol{X}_1, \boldsymbol{X}_2\right) &= \mathrm{MLP}_2(\boldsymbol{u}) \end{aligned} \tag{6}$$

where concatenation can retain all the information [54]. The elementwise product is a certain measure of "similarity" between two sentences [55]. Their differences can capture the degree of distributional inclusion in each dimension [56]. $\hat{y} \in \{1, 0\}$ indicates whether two input sequences have the same relation.

2) Triplet Distance Calculation: Apart from leveraging R^2 task to learn interclass relation information, we also intend to learn intraclass connections from triplet-based relation. Thus, we introduce a triplet loss [57] into R^2-Net. As a fundamental similarity function, a triplet loss is widely applied in information retrieval [7] and is able to pull together input sequences with the same label and push apart these with different labels. The inputs of this component are three semantic representations: $\boldsymbol{v}_a$ for anchor text $\boldsymbol{X}_a$, $\boldsymbol{v}_p$ for positive text $\boldsymbol{X}_p$, and $\boldsymbol{v}_n$ for negative text $\boldsymbol{X}_n$. We first transform them

into a common space with a full connection layer [44]. Then, we calculate the distance between anchor and positive, as well as the distance between anchor and negative

$$\begin{aligned} \bar{\boldsymbol{v}}_i &= \mathrm{ReLu}(\boldsymbol{W}_d \boldsymbol{v}_i + \boldsymbol{b}_d), \quad i \in \{a, p, n\} \\ d_{ap} &= \mathrm{Dist}\big(\bar{\boldsymbol{v}}_a, \bar{\boldsymbol{v}}_p\big), \quad d_{an} = \mathrm{Dist}\big(\bar{\boldsymbol{v}}_a, \bar{\boldsymbol{v}}_n\big) \end{aligned} \tag{7}$$

where $\{\boldsymbol{W}_d \in \mathcal{R}^{d_m * d_p}, \boldsymbol{b}_d \in \mathcal{R}^{d_m}\}$ are trainable parameters, d_m is the hidden state size, and $\mathrm{Dist}(\cdot)$ is the distance calculation function, which we use Euclidean distance.

C. Model Learning

As mentioned in Section III, both text classification and R^2 task can be treated as classification tasks. Thus, we employ cross entropy as the loss for each input as follows:

$$\begin{aligned} L_s &= -\sum_{i=1}^{N} \boldsymbol{y}_i \log P(y_i|\boldsymbol{X}_i) \\ L_{R^2} &= -\sum_{j=1}^{N/2} \hat{\boldsymbol{y}}_j \log P\big(\hat{y}_j|(\boldsymbol{X}_1, \boldsymbol{X}_2)_j\big) \end{aligned} \tag{8}$$

where N is the number of one training batch, $\boldsymbol{y}_i$ is the one-hot vector for the true label of the ith instance, and $\hat{\boldsymbol{y}}_j$ is the one-hot vector for the true R^2 of the jth instance pair.

Moreover, we introduce a triplet loss to better analyze the connections and differences among labels of different pairs

$$L_d = \sum_{i=1}^{N/3} \max\big(\big(d_{ap} - d_{an} + \mathrm{margin}\big)_i, 0\big). \tag{9}$$

Finally, we treat the weighted sum of these losses with a hyperparameter η as the loss function for one mini-batch

$$\mathrm{Loss} = \eta L_s + (1 - \eta)\big(L_{\mathrm{R}^2} + L_d\big). \tag{10}$$

V. Description-Enhanced Label Embedding Network

In our preliminary work [20], we propose R^2-Net to exploit label information in a one-hot manner for better supervision and text classification. However, one-hot usage still needs to improve in exploiting labels comprehensively. As mentioned in Section I, representing labels with dense vectors can be an inspiring direction. However, the particular meaning required in label semantics and multiple semantics contained in label words or phrases pose a big challenge for label embedding methods. Thus, in this section, we focus on describing and representing label semantics in a fine-grained manner.

Specifically, we propose to employ fine-grained descriptions to generate better label embedding and extend R^2-Net to a novel DELE. The overall architecture is reported in Fig. 3. The key contributions lie in the following two areas. The first is incorporating fine-grained descriptions from WordNet to enrich the label embedding learning process so that label semantics can be fully exploited. The second is developing a novel mutual interaction module based on the CL framework, which is used to analyze bidirectional interactions between input text sequences and labels. Along this line, label semantics can better guide the importance selection from input text.

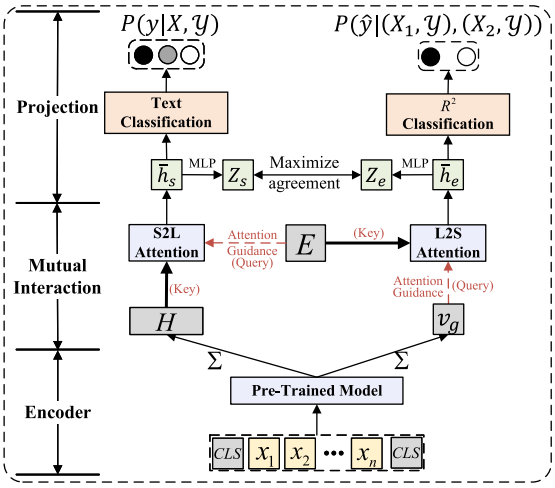


Fig. 3. Overall structure of DELE, which consists of encoder, our proposed mutual interaction, and projection. (Query) and (Key) denote the components in attention calculation.

Meanwhile, text representations can be used to denoise fine-grained descriptions for better label embedding learning.

As reported in Fig. 3, DELE consists of three modules: 1) encoder module: encoding input text sequence and all labels; 2) mutual interaction module: developing sentences to labels (S2L) attention and label to sentences (L2S) attention to enhance embedding learning and denoise fine-grained descriptions; and 3) projection module: projecting different learned embeddings into the same space for CL. Next, we will report the technical details of DELE.

A. Encoder

For input text sequence, we employ a similar operation to R^2-Net to encode the input text sequence and obtain the word representations $\boldsymbol{H}$ and global sequence representation $\boldsymbol{v}_g$.

As for labels, we design a novel label encoder to incorporate fine-grained descriptions for label embedding learning. Specifically, we employ WordNet as prior knowledge and retrieve relevant descriptions for labels. Since noun senses of words are usually employed as labels, we select noun descriptions of labels from WordNet. Moreover, words in WordNet have about 25 attributes (e.g., noun.act, noun.animal, and noun.event).[2] To ensure the relevance and mitigate the noise problem, we manually select no more than three noun descriptions for each label word based on most relevant attributes. If the label consists of more than one word (e.g., "Business & Finance" in Yahoo! Answer dataset), we select no more than three noun descriptions for each word and treat all of them as fine-grained descriptions for the label. After that, our designed label encoder is used to generate the corresponding representation.

Fig. 4 shows the structure of label encoder. Taking the ith label as an example, we first use $\boldsymbol{E}_f \in \mathbb{R}^{m * d_p}$ to represent the vanilla label embedding matrix. The embedding of the ith label can be obtained by extracting the ith column $\hat{\boldsymbol{e}}_i$ from $\boldsymbol{E}_f$. Meanwhile, we obtain fine-grained description set $\boldsymbol{S}^{(i)}$ for the ith label, where the jth description can be denoted as $\boldsymbol{s}_j^{(i)}$. Then, we employ BERT to encode these descriptions

[2]https://wordnet.princeton.edu/documentation/lexnames5wn

and average the output of "[CLS]" from the last layer as the description representation. Next, we add the vanilla embedding $\hat{\boldsymbol{e}}_i$ and the description representation $\bar{\boldsymbol{e}}_i$ to get the enriched representation $\boldsymbol{e}_i$ for the ith label as follows:

$$\begin{aligned}\bar{\boldsymbol{e}}_i &= \frac{1}{|\boldsymbol{S}^{(i)}|}\sum_{j=1}^{|\boldsymbol{S}^{(i)}|}\mathrm{BERT}_L\left(\boldsymbol{s}_j^{(i)}\right)\\ \hat{\boldsymbol{e}}_i &= \boldsymbol{y}_i\boldsymbol{E}_f, \quad \boldsymbol{e}_i = \hat{\boldsymbol{e}}_i + \bar{\boldsymbol{e}}_i.\end{aligned} \tag{11}$$

B. Mutual Interaction

In the previous part, we introduced fine-grained descriptions from WordNet to enrich the label embedding learning. However, only the specific meaning of a word is treated as a label. Thus, not all descriptions are helpful, and some even obscure label semantics. To this end, we develop a novel mutual interaction module based on CL to mitigate this unexpected noise. As shown in Fig. 3, this module leverages S2L attention and L2S attention to tackle this problem.

1) S2L Attention: Since label semantic meaning is highly related to input text, we first leverage text representation $\boldsymbol{v}_g$ as the guidance to select necessary label representations from label embedding $\boldsymbol{E}$ as follows:

$$\begin{aligned}\boldsymbol{E} &= [e_1, e_2, \ldots, e_m]\\ \boldsymbol{\beta}^l &= \boldsymbol{\omega}_l^T\tanh\left(\boldsymbol{W}_l\boldsymbol{E} + \boldsymbol{U}_l\boldsymbol{v}_g \otimes \boldsymbol{I}_l\right)\\ \bar{\boldsymbol{h}}_e &= \sum_{i=1}^{m}\frac{\exp\left(\beta_i^l\right)}{\sum_{k=1}^{m}\exp\left(\beta_k^l\right)}e_i\end{aligned} \tag{12}$$

where $\{\boldsymbol{\omega}_l \in \mathcal{R}^{1*d_a}, \boldsymbol{W}_l \in \mathcal{R}^{d_a*d_p}, \boldsymbol{U}_l \in \mathcal{R}^{d_a*d_p}\}$ are trainable parameters. d_a is the size of attention unit. $\boldsymbol{I}_l \in \mathbb{R}^m$ is a row vector of 1. $\boldsymbol{U}_l\boldsymbol{v}_g \otimes \boldsymbol{I}_l$ means repeating $\boldsymbol{U}_l\boldsymbol{v}_g$ m times. $\boldsymbol{\beta}^l$ is the unnormalized attention weight for label representations. $\bar{\boldsymbol{h}}_e$ denotes the text supervised semantic vector generated from label semantics. With this operation, the unexpected noise from fine-grained descriptions can be effectively mitigated and label semantics can be expressed accurately.

2) L2S Attention: Meanwhile, we intend to analyze the impact from label to text in a similar way. Since there are multiple labels, we adopt each label to guide the selection of input text. Then, max pooling is employed to merge results for label supervised semantic vectors

$$\begin{aligned}\beta^t &= \boldsymbol{\omega}^T\tanh(\boldsymbol{W}\boldsymbol{H} + \boldsymbol{U}\boldsymbol{e}_t \otimes \boldsymbol{I})\\ \bar{\boldsymbol{h}}_s^t &= \sum_{i=1}^{n}\frac{\exp\left(\beta_i^t\right)}{\sum_{k=1}^{n}\exp\left(\beta_k^t\right)}\boldsymbol{h}_i, \quad \boldsymbol{h}_i \in \boldsymbol{H}\\ \bar{\boldsymbol{h}}_s &= \mathrm{maxpooling}\left(\left[\bar{\boldsymbol{h}}_s^1, \bar{\boldsymbol{h}}_s^2, \ldots, \bar{\boldsymbol{h}}_s^m\right]\right)\end{aligned} \tag{13}$$

where $t \in \{1, 2, \ldots, m\}$ indicates the index of the tth label. β^t is the unnormalized attention weight for word representations under the guidance of the tth label. $\{\boldsymbol{\omega} \in \mathcal{R}^{1*L_s}, \boldsymbol{W} \in \mathcal{R}^{L_s*d_p}, \boldsymbol{U} \in \mathcal{R}^{L_s*d_p}\}$ are also trainable parameters. L_s is the sentence length. $\bar{\boldsymbol{h}}_s$ denotes the label supervised semantic vector from sentence semantics.

C. Projection

In Sections V-A and V-B, we have obtained semantic vectors from the text perspective and label perspective. To minimize their distance for better representation learning and classification, inspired by SimCLR [58], we employ an MLP to project the semantic vectors into the same space for quality improvement of learned semantic vectors as follows:

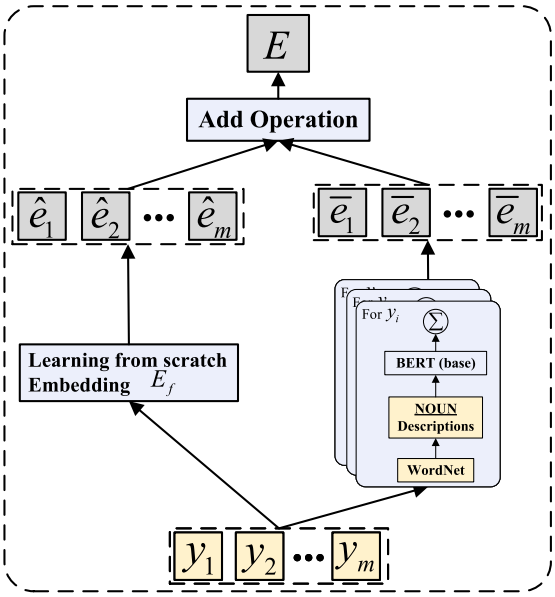


Fig. 4. Architecture of label encoder in DELE, which incorporates vanilla embedding and our proposed description-enhanced embedding.

$$\boldsymbol{z}_e = \mathrm{MLP}_2\left(\bar{\boldsymbol{h}}_e\right), \quad \boldsymbol{z}_s = \mathrm{MLP}_2\left(\bar{\boldsymbol{h}}_s\right) \tag{14}$$

where $\boldsymbol{z}_e$ and $\boldsymbol{z}_s$ are projection vectors for the contrastive loss. Since our target is to predict the most suitable label for input text and $\boldsymbol{h}_s$ is generated from the text perspective, we select label supervised semantic vector $\bar{\boldsymbol{h}}_s$ to make the final decision:

$$\begin{aligned}P(y|\boldsymbol{X}, \mathcal{Y}) &= \mathrm{MLP}_3\left(\bar{\boldsymbol{h}}_s\right)\\ y^* &= \mathrm{argmax}_{y\in\mathcal{Y}}P(\boldsymbol{y}|\boldsymbol{X}, \mathcal{Y}).\end{aligned} \tag{15}$$

Similarly, we also employ the R^2 classification task to assist in the improvement of model ability. A text supervised semantic vector $\boldsymbol{h}_e$ is selected to make the prediction as follows:

$$\begin{aligned}\hat{\boldsymbol{u}} &= \left[\bar{\boldsymbol{h}}_e^1; \bar{\boldsymbol{h}}_e^2; \left(\bar{\boldsymbol{h}}_e^1 \odot \bar{\boldsymbol{h}}_e^2\right); \left(\bar{\boldsymbol{h}}_e^1 - \bar{\boldsymbol{h}}_e^2\right)\right]\\ P\left(\hat{y}|(\boldsymbol{X}_1, \mathcal{Y}), (\boldsymbol{X}_2, \mathcal{Y})\right) &= \mathrm{MLP}_4\left(\hat{\boldsymbol{u}}\right)\\ \hat{y}^* &= \mathrm{argmax}_{\hat{y}\in\{0,1\}}P\left(\hat{y}|(\boldsymbol{X}_1, \mathcal{Y}), (\boldsymbol{X}_2, \mathcal{Y})\right)\end{aligned} \tag{16}$$

where $\mathrm{MLP}_3(\cdot)$ and $\mathrm{MLP}_4(\cdot)$ are MLPs, which consist of one hidden layer and a softmax output layer, and y^* and $\hat{y}^*$ are the predicted label for $\boldsymbol{X}$ and predicted R^2 of $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$.

D. Model Learning

DELE has three targets to optimize, including CL target, R^2 classification target, and text classification target. They have been listed in the following.

1) CL Target: As mentioned before, minimizing the distance between semantic representations of input text and the proper label is pivotal for text classification. Thus, we select NT-Xent [58] as the contrastive loss, where the positive pair consists of label supervised semantic vector and text

supervised semantic vector, and negative examples are the rest instances in the current batch

$$L_1 = \sum_{i=1}^{N} -\log \frac{\text{sim}\left(\boldsymbol{z}_s^i, \boldsymbol{z}_e^i\right)/\tau}{\sum_{j=1}^{K} \mathbb{1}_{[j\neq i]} \text{sim}\left(\boldsymbol{z}_s^i, \boldsymbol{z}_s^j\right)/\tau} \tag{17}$$

where $\text{sim}(\cdot)$ is the similarity calculation function, N is the number of one training batch, and τ is the temperature parameter. $\mathbb{1}_{[j\neq i]}$ is an indicator function evaluating to 1 iff $j \neq i$.

2) R^2 Classification Target: For this classification target, we select cross entropy as the training loss

$$L_2 = \sum_{i=1}^{N/2} -\hat{\boldsymbol{y}}_i \log P\left(\hat{y}_i | (\boldsymbol{X}_1, \mathcal{Y})_i, (\boldsymbol{X}_2, \mathcal{Y})_i\right) \tag{18}$$

where $\hat{\boldsymbol{y}}_i$ is the one-hot ground truth that indicates the true R^2 of the ith pair.

3) Text Classification Target: For this task-related target, we select cross entropy as the training loss

$$L_3 = \sum_{i=1}^{N} -\boldsymbol{y}_i \log P(y_i | \boldsymbol{X}, \mathcal{Y}) \tag{19}$$

where $\boldsymbol{y}_i$ is the one-hot ground truth that represents the true label of the ith example.

The final loss for one mini-batch can be calculated with a weighted sum operation. Here, we leverage weight δ and μ to control the impacts of CL target and R^2 task for the final classification performance

$$\text{Loss} = \delta L_1 + \mu L_2 + L_3. \tag{20}$$

VI. Experiments

In this section, the evaluation datasets and metrics are first introduced. Then, model implementation and training details are reported for better illustration. Next, empirical results, a detailed analysis of models, and experimental results are presented. For all reported results, we employ boldface and underline for the best and the second-best results, respectively.

A. Datasets and Evaluation Method

To evaluate our proposed R^2-Net and DELE comprehensively, we select different benchmark datasets: *SNLI* [25], *SICK* [63], and *SciTail* [64] for NLI, *Quora Question Pair (Quora)* [26] and *MSRP* [1] datasets for PI, *Yahoo! Answers (Yahoo)* for QA topic classification, as well as *SST-5* [27] for sentiment classification. These tasks focus on different aspects and exhibit different characteristics of text classification task.

For evaluation, we select accuracy and error rate comparison as evaluation metrics, which are the same as most baselines did. We have to note that for each experiment, we repeat the evaluation process five times with different seeds and random initialization and report the best results for our proposed methods and baselines.

TABLE II

Hyperparameters Configuration in R^2-Net and DELE

Model	Hyper-parameters	Value
BERT-base	12 layers, hidden size $d_p = 768$, attention heads 12	
Roberta-base	12 layers, hidden size $d_p = 768$, attention heads 12	
R^2-Net	Kernel size of *CNN*	$d_k = \{1, 2, 3\}$
	hidden size of MLP_1	$d_m = 300$
	hidden size of MLP_2	$d_2 = 300$
	Backbone learning rate	$lr_1 = 10^{-5}$
	The other learning rate	$lr_2 = 10^{-3}$
	margin in loss L_d	$margin = 0.2$
DELE	number of backbone layer	$L = 2$
	attention size of mutual interaction	$d_a = 100$
	hidden size of MLP_3 and MLP_4	$d_3 = 200$

B. Model Implementation and Training Details

To obtain the best performance, we have tuned hyperparameters on validation set of each dataset and used early stop operation to select the best values for hyperparameters. BERT-(base) and Roberta-(base) are selected as backbones. We have to note that all baselines have the same experimental settings for a fair comparison. In order to achieve the best model performance, R^2-Net and DELE have different parameter settings over different datasets. Therefore, we report common hyperparameters in Table II and summarize them here.

For R^2-Net, kernel sizes of CNN in local encoding are $d_k = 1, 2$, and 3. The hidden size of MLP_1 is $d_m = 300$. The margin in (9) is margin $= 0.2$. For PLMs, we set the learning rate of 10^{-5} and use AdamW to fine-tune parameters. For the rest parameters, the learning rate is set as 10^{-3} and decreases as the model training. An Adam optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.999$ is adopted to optimize these parameters.

For DELE, the number of used output layers in PLMs is $L = 2$. The attention size in the mutual interaction module is $d_a = 100$. The hidden size of MLP_2 in projection is $d_2 = 300$. The hidden size of MLP_3 and MLP_4 the in classification layer is $d_3 = 200$. For model training, we leverage Adam as the optimizer. Inspired by [65], we develop the following operation to control the learning rate in the ith training batch:

$$\begin{aligned} \text{lr} &= \epsilon(\text{lr}_{\max} - \text{lr}_{\min}) + \text{lr}_{\min} \\ \mu &= \begin{cases} 0.5\cos\left(\dfrac{\pi}{\eta C}(i - \eta C + 1)\right) + 0.5, & \text{if } i \leq \eta C \\ \left(0.5\cos\left(\dfrac{\pi}{C - \eta C}(i - \eta C)\right) + 0.5\right)^2, & \text{if } i > \eta C \end{cases} \end{aligned} \tag{21}$$

where $\text{lr}_{\max} = 1$, $\text{lr}_{\min} = 0.000001$, C is the total number of training batches, and $\eta \in [0, 1]$ is the percentage of the warm-up loops.

C. Model Complexity Analysis

In order to better demonstrate the superiority of our proposed methods, we make an additional computational complexity analysis. Since our proposed methods are both based on PLMs, we only analyze the extra computational complexity. For our preliminary work R^2-Net, additional components consist of CNN-based local encoder, R^2 classification, and triplet distance calculation components. For the CNN-based local

TABLE III
EXPERIMENTAL RESULTS (ACCURACY) ON DIFFERENT DATASETS FROM NLI TASK. + AND − DENOTE THE PERCENTAGE OF DECREASE OR INCREASE IN ERROR RATE COMPARED WITH BACKBONES

Type	Model	SNLI Full Test (3-classes)	SNLI Hard Test (3-classes)	SICK Test (3-classes)	SciTail Test (2-classes)
One-hot Encoding	(1) DRCN [2]	86.5%	68.3%	87.4%	85.7%
	(2) CSRAN [59]	88.5%	76.8%	89.7%	86.5%
	(3) RE2 [60]	88.9%	77.3%	89.8%	86.2%
	(4) DRr-Net [10]	87.7%	71.4%	88.3%	87.4%
	(5) SimCSE [47]	89.4%	81.2%	90.3%	93.3%
	(6) BERT-(base) [21]	90.3% (0.0%)	80.6% (0.0%)	88.7% (0.0%)	93.1% (0.0%)
	(7) BERT-(large) [21]	90.7% (+4.12%)	81.3% (+3.61%)	88.3% (−3.54%)	93.6% (+7.25%)
	(8) RoBERTa-(base) [28]	90.9% (0.0%)	81.5% (0.0%)	90.3% (0.0%)	93.8% (0.0%)
	(9) ALBERT-(base) [61]	86.2%	77.5%	87.3%	91.4%
Label Embedding	(10)FLE-BERT(base) [15]	90.5% (+2.06%)	80.4% (−1.03%)	89.3% (+5.31%)	93.4% (+4.35%)
	(11) LEAM-BERT(base) [62]	90.3% (0.0%)	80.8% (+1.03%)	89.0% (+2.65%)	93.4% (+4.35%)
	(13) EXAM-BERT(base) [36]	90.6% (+3.09%)	81.0% (+2.06%)	89.5% (+7.08%)	93.8% (+10.14%)
	(12) LGDSC-BERT(base) [35]	91.3% (+10.31%)	81.4% (+4.12%)	89.5% (+7.08%)	94.0% (+13.04%)
	(14) LCM-BERT(base) [14]	90.8% (+5.15%)	81.3% (+3.61%)	90.3% (+14.16%)	94.1% (+14.49%)
	(15) EXAM-RoBERTa(base) [36]	91.5% (+6.59%)	81.9% (+2.16%)	90.5% (+2.06%)	94.1% (+4.84%)
Our Methods	(16) *R^2-Net*-BERT-(base)	91.1% (+8.25%)	81.0% (+2.06%)	89.2% (+4.42%)	92.9% (−2.89%)
	(17) *R^2-Net*-RoBERTa-(base)	91.3% (+4.40%)	81.4% (−0.54%)	89.5% (−8.24)	93.9% (+1.61)
	(18) *DELE*-BERT-(base)	91.3% (+10.31%)	81.6% (+5.15%)	89.9% (+10.16%)	94.1% (+14.49%)
	(19) *DELE*-RoBERTa-(base)	**91.8%** (+9.89%)	**83.2%** (+9.19%)	**90.7%** (+4.12%)	**94.5%** (+11.29%)

encoder, we leverage three kernels with size $d_k = \{1, 2, 3\}$ and an MLP. The extra parameter size is $14 + d_p * (6 * d_p)$. For the rest of two components, three different MLPs are used. Therefore, the extra parameter size is $(d_m * 2d_p) + (d_p * 4d_m) + (d_m * d_p)$. Therefore, the total extra computational complexity increase is $(7d_p + 6d_m) * d_p + 14$.

For our proposed DELE, additional components include mutual interaction module and R^2 classification layer. For the former, the extra parameter size is $(d_a + d_a * d_p + d_a * d_p) + (L_s + L_s * d_p + L_s * d_p)$. For the latter, the extra parameter size is $d_4 * 4d_p + d_4$. To this end, the total extra computational complexity increase is $(2d_a + 2L_s + 4d_4) * d_p + (d_a + L_s + d_4)$. Note that L_s is the sentence length and other notation values can be found in Table II. For our proposed label encoder, since we leverage the same PLMs to encode the fine-grained descriptions as input encoder, this label encoder does not add additional computational complexity.

As reported in Table II, the extra computational complexity in our proposed methods is acceptable in practice. Moreover, in DELE, we make full use of label information to generate positive semantic representations for input sentences and use in-batch negative sampling to obtain the negative semantic representations for each input sentence. Furthermore, to alleviate the computational complexity of pairwise CL, we leverage the same method as in [47], which transfers the pairwise calculation into a cross-entropy calculation during implementation. This operation can largely decrease the computational complexity of pairwise CL.

D. Overall Experimental Results

In this section, we will give a detailed analysis of experimental results by reporting accuracy and error rate comparison with backbones. Note that the error rate comparison is based on the corresponding backbone. For example, R^2-Net-BERT-(base) achieves +8.25% improvement in Table III(16), which denotes that it decreases the error rate by 8.25% compared with backbone BERT-(base). The detailed analysis is reported in the following.

1) Performance on NLI Task: Table III reports the results on the NLI task. We can conclude that our proposed R^2-Net and DELE achieve highly comparable performance over all datasets. Moreover, DELE-RoBERTa-(base) has the best performance. To achieve such advantages, we first select PLMs as backbones so that knowledge from large corpora can be accessed. This is one of the reasons that our methods outperform other baselines, especially better than these PLM-free baselines [Table III(1)–(4)] by a large margin. Second, we develop a novel R^2 task to help models fully exploit label information in a one-hot manner. Along this line, R^2-Net and DELE can obtain intraclass and interclass knowledge among input texts with the same or different labels, which is in favor of achieving better performance than all baselines, including PLM baselines [Table III(5)–(9)]. Moreover, we design to incorporate fine-grained descriptions from WordNet into label embedding learning so that particular label semantics can be better represented. This is another vital reason that DELE achieves the best performance, which also demonstrates the effectiveness of our proposed description-enhanced label embedding method.

Moreover, incorporating label information (one-hot encoding or label embedding) can better ensure model performance when dealing with difficult examples (hard test in Table III), in which text sequences with obviously identical words have been removed [17]. By employing R^2 task to measure labels in a one-hot manner, R^2-Net can better exploit implicit information from the input text, which leads to an inspiring performance. Moreover, label embedding methods can exploit label information better than one-hot encoding methods. Therefore, we observe the superiority of label embedding methods. Furthermore, DELE employs fine-grained descriptions to enhance label embedding learning. Thus, it further improves the performance of label embedding methods and achieves the best performance.

To make a fair comparison, we replace the basic encoders of label embedding methods (e.g., LEAM and EXAM) with the same PLMs backbones as our methods did and report results

TABLE IV

RESULTS (ACCURACY) ON PI, SENTIMENT ANALYSIS, AND QA TOPIC CLASSIFICATION TASK. + AND − DENOTE THE PERCENTAGE OF DECREASE OR INCREASE IN ERROR RATE COMPARED WITH BACKBONES

Type	Model	Quora Test (2-classes)	MSRP Test (2-classes)	STS-5 Test (5-classes)	Yahoo! Answer Test (10-classes)
One-hot Encoding	(1) DRCN [2]	90.2%	82.5%	-	75.1%
	(2) RE2 [60]	89.3%	78.5%	-	75.5%
	(3) DRr-Net [10]	89.8%	82.9%	50.8%	74.7%
	(4) SimCSE [47]	91.5%	84.8%	54.1%	76.2%
	(5) BERT-(base) [21]	91.0% (0.0%)	84.2% (0.0%)	53.1% (0.0%)	75.7% (0.0%)
	(6) BERT-(large) [21]	91.4% (+4.44%)	85.4% (+7.59%)	54.9% (+3.84%)	76.3% (+2.67%)
	(7) RoBERTa-(base) [28]	90.6% (0.0%)	87.1% (0.0%)	56.2% (0.0%)	75.9% (0.0%)
	(8) ALBERT-(base) [61]	90.3%	**88.6%**	47.3%	74.2%
Label Embedding	(9)FLE-BERT(base) [15]	91.2% (+2.22%)	84.2% (0.0%)	53.4% (+0.64%)	75.6% (−0.41%)
	(10) LEAM-BERT(base) [62]	91.3% (+3.34%)	83.9% (−1.90%)	53.8% (+1.49%)	75.9% (+0.82%)
	(11) EXAM-BERT(base) [36]	91.4% (+4.44%)	84.2% (0.0%)	54.3% (+2.56%)	76.3% (+2.47%)
	(12) LGDSC-BERT(base) [35]	91.6% (+6.67%)	84.5% (+1.90%)	54.3% (+2.56%)	76.2% (+2.06%)
	(13) LCM-BERT(base) [14]	91.5% (+5.56%)	84.2% (0.0%)	54.6% (+3.20%)	76.4% (+2.88%)
	(14) EXAM-RoBERTa(base) [36]	92.0% (+14.89%)	85.6% (−11.63%)	57.0% (+1.82%)	76.3% (+1.65%)
Our Methods	(15) R^2-*Net*-BERT-(base)	91.6% (+6.67%)	84.3% (+0.63%)	54.1% (+2.13%)	76.2% (+2.06%)
	(16) R^2-*Net*-RoBERTa-(base)	91.7% (+11.70%)	84.5% (−20.15%)	56.9% (+1.60%)	76.3% (+1.65%)
	(17) *DELE*-BERT-(base)	91.7% (+7.78%)	84.8% (+3.80%)	54.7% (+3.41%)	76.5% (+3.29%)
	(18) *DELE*-RoBERTa-(base)	**92.3%** (+18.09%)	86.7% (−3.10%)	**57.8%** (+3.65%)	**76.8%** (+3.73%)

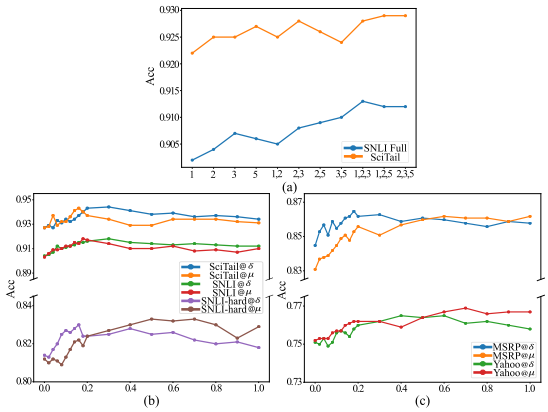


Fig. 5. Results of R^2-Net with different kernel sizes and DELE with different σ and μ on different datasets. (a) Results with different kernel size settings. (b) Results with different and on NLI task. (c) Results with different and on PI and QA TC task.

in Table III(10)–(15), in which DELE still has a better performance. The advantages of DELE lie in the more advanced label encoder and the consideration of denoising operation for additional knowledge. For label encoder, DELE not only learns the embedding during the model training so that the learned results are suitable for the current task but also employs multiaspect real sentences from WordNet as descriptions to enhance label semantic modeling. Meanwhile, considering that introducing fine-grained descriptions will import unexpected noise, we develop a novel mutual interaction module based on CL to leverage the attention mechanism to select the most relevant parts from the input text and labels simultaneously. Then, with the help of CL framework, DELE is able to use proper fine-grained knowledge to enhance the understanding of labels and improve the model performance.

2) Performance on PI Task: Apart from the NLI task, we also select the PI task to evaluate the model performance. PI task concerns whether two sentences express the same meaning and has broad applications in QA communities.[3] Table IV reports the corresponding results. We observe that R^2-Net and DELE still achieve highly competitive performance over other baselines. For one thing, the results demonstrate that though the PI task is a binary classification task, its labels still contain some useful semantics. For another thing, we can conclude that our proposed R^2 task and label embedding methods with fine-grained descriptions are useful for exploiting label information and boosting model performance.

Besides, we obtain that almost all models have a better performance on Quora than MSRP. One possible reason is that Quora has more data (over 400k sentence pairs in Quora versus 5801 sentence pairs in MSRP). Besides, intersentence interaction is probably another reason. Lan and Xu [66] observed that the Quora dataset contains many sentence pairs with less complicated interactions (many identical words in sentence pairs). Therefore, we can obtain that ALBERT achieves the best performance on the MSRP dataset.

3) Performance on Sentiment Analysis and QA Topic Classification: To make a better evaluate, we further conduct experiments on *SST-5* and *Yahoo! Answer* datasets, which have more complex and realistic labels (e.g., *somewhat positive, Business*). Table IV summarizes the results. Similarly, our proposed methods achieve stable and comparable performance. Moreover, different from previous experiments, label embedding baselines (i.e., EXAM and LGDSC) do not achieve a better improvement, compared with PLMs baselines. One possible reason is that embedding these complicated labels with coarse-grained information cannot capture sufficient useful information for label exploitation and text classification.

Moreover, compared with PLMs baselines, the improvement of DELE is not so obvious. After analyzing the dataset deeply, we observe that some QA pairs in this dataset have more than 512 words, as well as irregular textual expressions. These input noises will do harm to label utilization since we leverage text representations to guide the denoising process of label

[3]https://www.quora.com/

TABLE V
ABLATION PERFORMANCE (ACCURACY) OF R^2-NET-BERT-(BASE)

Model	SNLI Full Test	SciTail Test
(1) BERT-(base)	90.3%	92.0%
(2) *R^2-Net* (w/o local encoder)	90.7%	92.6%
(3) *R^2-Net* (w/o R^2 task learning)	90.5%	92.3%
(4) *R^2-Net* (w/o triplet loss)	90.9%	92.6%
(5) *R^2-Net* (w/ NT-Xent loss)	**91.3%**	**93.3%**
(6) *R^2-Net*-BERT-(base)	91.1%	92.9%

TABLE VI
ABLATION PERFORMANCE (ACCURACY) OF DELE-BERT-(BASE)

Model	SNLI Full Test	SciTail Test
(1) BERT-(base)	90.3%	93.1%
(2) BERT + Atten	90.6%	93.5%
(3) $\bar{\boldsymbol{h}}_s$(w/o mutual interaction)	90.4%	93.2%
(4) $\bar{\boldsymbol{h}}_e$(w/o mutual interaction)	*43.7%*	*65.4%*
(5) $\bar{\boldsymbol{h}}_s$(w/o description)	90.8%	93.8%
(6) $\bar{\boldsymbol{h}}_e$(w/o description)	89.3%	90.1%
(7) $\bar{\boldsymbol{h}}_s$(w/o R^2 task))	90.3%	92.9%
(8) $\bar{\boldsymbol{h}}_e$(w/o R^2 task)	88.5%	90.4%
(9) *DELE*-BERT-(base)	**91.3%**	**94.1%**

description utilization. Meanwhile, we only select no more than three descriptions manually, which may be insufficient for those labels that have a wealth of semantics. Furthermore, the particular meaning of label words requires a precise selection of label descriptions. However, DELE leverages the attention mechanism to select the most relevant parts from label embedding $\boldsymbol{E}$, where vanilla embedding and description embedding have already been integrated. Applying denoising operation at an earlier stage may achieve better performance. Nevertheless, on the contrary, DELE has made an early attempt at leveraging fine-grained knowledge to enhance label embedding and denoising of knowledge utilization, which also illustrates the advancement of DELE.

4) Parameter Sensitive Test: As shown in (4) and (20), kernel sizes d_k in R^2-Netand $\{\delta, \mu\}$ in DELE are essential for model performance. To this end, we conduct additional parameter sensitive test to verify their impact.

For kernel sizes d_k, it aims to extract the local information of input text at different scales. Results with different kernel size settings are reported in Fig. 5(a). We can observe that R^2-Net will have better performance with more kernel sizes, in which $\{1, 2, 3\}$, $\{1, 2, 5\}$, and $\{2, 3, 5\}$ achieve the best performance. Moreover, using $\{1, 2, 3\}$ as kernel sizes has more stable performance. We speculate that the possible reason is that these kernel sizes can pay more attention to uni-gram, bi-gram, as well as tri-gram information, which is helpful for enhancing the representation learning of PLMs. Meanwhile, using $\{2, 3, 5\}$ as kernel sizes has similar performance, which also demonstrates its effectiveness.

For σ and μ, when removing CL or R^2 task (i.e., $\delta = 0$ or $\mu = 0$), model performance cannot be comparable with the performance with CL or R^2 task (i.e., $\delta \neq 0$ or $\mu \neq 0$), demonstrating the necessity of these two SSL frameworks. Moreover, with the increase in label size or label semantics, the best values of δ and μ also become larger. For one thing, this phenomenon proves that our proposed SSL can extract more critical information from more diverse and realistic labels and have a bigger impact on model performance, proving the necessity of better label utilization methods. Furthermore, we observe that μ has a bigger impact than α when labels are harder, which is somewhat counterintuitive and different from other results. The possible reason is the underutilization of label information. Limited information and unexpected noise will decrease the effectiveness of our proposed CL framework. This finding is similar to the observation in Section VI-D3.

5) Ablation Study: The overall experiments have proved the superiority of our proposed methods. However, which parts in R^2-Net and DELE play a more important role in label utilization and performance improvement is still unclear. Therefore, we perform an ablation study to conduct a comprehensive analysis. The corresponding results are reported in Tables V and VI. Note that we select BERT-(base) as the backbone to compare the importance of each part. Leveraging RoBERTa-(base) as the backbone will obtain similar results.

First, for R^2-Net, we conduct experiments on CNN-based local encoder, R^2 task classification, as well as loss function. Results are summarized in Table V. According to the results, we can observe varying degrees of model performance decline. Among all of them, the R^2 task has the biggest impact, and triple loss has a relatively small impact on the model performance. Replacing triplet loss with contrastive loss will improve model performance slightly. These observations prove that the R^2 task is more important for relation information utilization. R^2 task and better contrastive loss will further improve the model performance.

Second, for DELE, we focus on label embedding module, mutual interaction module, and R^2 task. Results are reported in Table VI. Here, "BERT + Atten" denotes leveraging vanilla label embedding to select important parts of text without CL "$\bar{\boldsymbol{h}}_s$ (w/o mutual interaction)" indicates leveraging $\bar{\boldsymbol{h}}_s$ for classification and no interactions between input text and labels. Other ablation experiments have similar settings. According to the results, we obtain some observations. First, results (3) and (4) demonstrate the necessity of mutual interactions between input text and labels, especially result (4). When there is no interaction between input text and labels, label semantics cannot handle final classification at all. Second, results (5) and (6) prove that utilizing fine-grained descriptions is helpful for semantic understanding and performance improvement. Moreover, results (7) and (8) inspire us that the R^2 task can provide additional signals for label utilization, which demonstrates the effectiveness and necessity of better label utilization methods. Last but not least, the comparison between $\bar{\boldsymbol{h}}_s$ and $\bar{\boldsymbol{h}}_e$ with the same setting (e.g., w/o description) indicates that CL can constrain learned representations from different views to have similar semantics. However, $\bar{\boldsymbol{h}}_e$ cannot be the replacement for input text, only an enhancement to input text.

6) Case Study of R^2-Net and DELE: To provide some intuitionistic examples for explaining the superiority of our models, we sample 1500 sentence pairs from SNLI Test and send them to BERT-(base) baseline, EXAM-BERT-(base) baseline, R^2-Net, and DELE to generate representations. Then, we use t-sne [67] to visualize them with the same parameter settings. Fig. 6 reports the corresponding results. By comparing

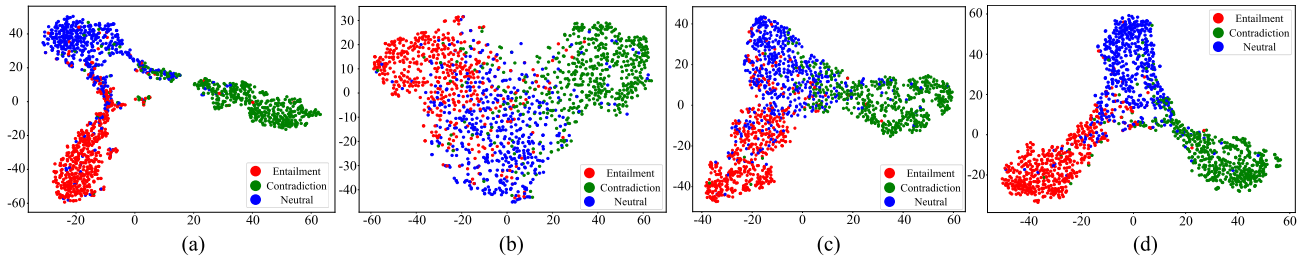


Fig. 6. Visualization of semantic representations from two representative baselines: (a) BERT-(base) and (b) EXAM, and our proposed (c) R^2-Netand (d) DELE. Note that BERT-(base) is selected as the backbone in this experiment.

Fig. 6(a)–(d), we observe that representations generated by DELE have closer intraclass distances and more distinguishable interclass distances. In other words, by considering fine-grained knowledge and mutual interactions, DELE is able to pull together samples with the same label and push apart samples with different labels, which is helpful for better text classification. These observations not only explain why DELE achieves impressive performance but also demonstrate the necessity of more detailed and cleaner label descriptions, as well as more comprehensive analysis between input text and labels. All of these are very helpful for label semantics utilization and text classification performance improvement.

VII. Discussion

Here, we discuss the impact of the future directions of this study. To fully exploit label information for text classification, we develop a novel R^2-Net and DELE from the one-hot encoding perspective and label embedding perspective separately. Despite the progress we have achieved, our work can also provide some inspiration for relative research. For example, in extreme multilabel classification (XMC) tasks, label embedding matrix learning is a challenging problem. Inappropriate embedding methods may perform worse than sparse one-vs-all and partitioning approaches in XMC. To this end, our proposed methods provide a novel strategy to generate accurate label embeddings, which may inspire relative research, such as using additional descriptions to distinguish the similarity relations among hundreds of labels.

Meanwhile, our proposed methods still have space for further improvement. For example, when selecting fine-grained descriptions as additional knowledge, we employ a manual approach to select relevant descriptions, which is insufficient for complex labels and impractical for XMC. Therefore, better knowledge selection methods or multimodal knowledge (e.g., relations and hierarchical structures among different labels) are needed to boost semantic representation learning further. When talking about additional information, we only consider label descriptions. The relations and hierarchical structures among labels are also beneficial for label semantic modeling. Thus, in the future, we plan to exploit the label relations in XMC and use graph neural network (GNN) to obtain relation-aware label embeddings. Then, we can transfer our proposed methods to XMC by incorporating relation-aware label embeddings with our proposed description-enhanced label embeddings for model performance enhancement.

VIII. Conclusion and Future Work

In this article, we argued that current text classification methods are deficient in label utilization and ignore the guidance and semantic information contained in labels. Then, we presented a study on the exploitation of labels from a one-hot encoding manner and label embedding manners. Specifically, inspired by SSL methods used in PLMs, we developed a novel R^2 task to mine the potential of labels from a one-hot encoding perspective. Based on this novel self-supervised task, we designed a simple but effective method named R^2-Net for text classification. Meanwhile, a triplet loss is employed to constrain R^2-Net for better label relation analysis. One step further, since a one-hot encoding method still has some weaknesses in exploiting label information, we focused on label embedding methods and proposed a novel DELE for better label utilization. In DELE, fine-grained descriptions from WordNet are adopted to complete the label embedding learning. Also, a mutual interaction module is designed to denoise the additional knowledge and select the most relevant parts from input text and labels simultaneously. Finally, extensive experiments on multiple benchmark datasets demonstrate the superiority of R^2-Net and DELE, as well as the usefulness of initial attempts on fine-grained knowledge utilization for label embedding.

References

[1] W. B. Dolan and C. Brockett, "Automatically constructing a corpus of sentential paraphrases," in *Proc. IWP*, 2005, pp. 1–8.
[2] S. Kim, J.-H. Hong, I. Kang, and N. Kwak, "Semantic sentence matching with densely-connected recurrent and co-attentive information," in *Proc. AAAI*, 2019, pp. 6586–6593.
[3] D. Chen, A. Fisch, J. Weston, and A. Bordes, "Reading Wikipedia to answer open-domain questions," in *Proc. ACL*, 2017, pp. 1870–1879.
[4] S. F. Yilmaz, E. B. Kaynak, A. Koç, H. Dibeklioglu, and S. S. Kozat, "Multi-label sentiment analysis on 100 languages with dynamic weighting for label imbalance," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 1, pp. 331–343, Jan. 2023.
[5] F. Huang, X. Li, C. Yuan, S. Zhang, J. Zhang, and S. Qiao, "Attention-emotion-enhanced convolutional LSTM for sentiment analysis," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 9, pp. 4332–4345, Sep. 2022.
[6] L. Zhu, W. Li, Y. Shi, and K. Guo, "SentiVec: Learning sentiment-context vector via kernel optimization function for sentiment analysis," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 6, pp. 2561–2572, Jun. 2021.
[7] T.-Y. Liu, "Learning to rank for information retrieval," *Found. Trends Inf. Retr.*, vol. 3, no. 3, pp. 225–331, 2009.
[8] Q. Liu et al., "Finding similar exercises in online education systems," in *Proc. SIGKDD*, Jul. 2018, pp. 1821–1830.
[9] I. V. Serban, A. Sordoni, Y. Bengio, A. C. Courville, and J. Pineau, "Building end-to-end dialogue systems using generative hierarchical neural network models," in *Proc. AAAI*, 2016, pp. 3776–3783.

[10] K. Zhang et al., "DRr-Net: Dynamic re-read network for sentence semantic matching," in *Proc. AAAI*, vol. 33, 2019, pp. 7442–7449.

[11] K. Zhang, G. Lv, L. Wu, E. Chen, Q. Liu, and M. Wang, "LadRa-Net: Locally aware dynamic reread attention net for sentence semantic matching," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 2, pp. 853–866, Feb. 2023.

[12] Z. Tan, J. Chen, Q. Kang, M. Zhou, A. Abusorrah, and K. Sedraoui, "Dynamic embedding projection-gated convolutional neural networks for text classification," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 3, pp. 973–982, Mar. 2022.

[13] K. Zhang et al., "Image-enhanced multi-level sentence representation net for natural language inference," in *Proc. IEEE ICDM*, Nov. 2018, pp. 747–756.

[14] B. Guo, S. Han, X. Han, H. Huang, and T. Lu, "Label confusion learning to enhance text classification models," 2020, *arXiv:2012.04987*.

[15] Y. Xiong, Y. Feng, H. Wu, H. Kamigaito, and M. Okumura, "Fusing label embedding into BERT: An efficient improvement for text classification," in *Proc. ACL-IJCNLP*, 2021, pp. 1743–1750.

[16] H. Zhang, L. Xiao, W. Chen, Y. Wang, and Y. Jin, "Multi-task label embedding for text classification," in *Proc. EMNLP*, 2018, pp. 4545–4553.

[17] S. Gururangan, S. Swayamdipta, O. Levy, R. Schwartz, S. Bowman, and N. A. Smith, "Annotation artifacts in natural language inference data," in *Proc. NAACL-HLT*, 2018, pp. 107–112.

[18] L. Xiao, X. Huang, B. Chen, and L. Jing, "Label-specific document representation for multi-label text classification," in *Proc. EMNLP-IJCNLP*, 2019, pp. 466–475.

[19] R. Zhang, Y.-S. Wang, Y. Yang, D. Yu, T. Vu, and L. Lei, "Long-tailed extreme multi-label text classification with generated pseudo label descriptions," 2022, *arXiv:2204.00958*.

[20] K. Zhang, L. Wu, G. Lv, M. Wang, E. Chen, and S. Ruan, "Making the relation matters: Relation of relation learning network for sentence semantic matching," in *Proc. AAAI*, 2021, pp. 14411–14419.

[21] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," 2018, *arXiv:1810.04805*.

[22] Y. Kim, "Convolutional neural networks for sentence classification," 2014, *arXiv:1408.5882*.

[23] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," 2014, *arXiv:1412.3555*.

[24] A. Parikh, O. Täckström, D. Das, and J. Uszkoreit, "A decomposable attention model for natural language inference," in *Proc. EMNLP*, 2016, pp. 2249–2255.

[25] S. R. Bowman, G. Angeli, C. Potts, and C. D. Manning, "A large annotated corpus for learning natural language inference," in *Proc. EMNLP*, 2015, pp. 632–642.

[26] S. Iyer, N. Dandekar, and K. Csernai, "First quora dataset release: Question pairs," Tech. Rep., 2017. [Online]. Available: https://quoradata.quora.com/First-Quora-Dataset-Release-Question-Pairs

[27] R. Socher et al., "Recursive deep models for semantic compositionality over a sentiment treebank," in *Proc. EMNLP*, 2013, pp. 1631–1642.

[28] Y. Liu et al., "RoBERTa: A robustly optimized BERT pretraining approach," 2019, *arXiv:1907.11692*.

[29] W. Xu and Y. Tan, "Semisupervised text classification by variational autoencoder," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 1, pp. 295–308, Jan. 2020.

[30] D. Zeng, K. Liu, S. Lai, G. Zhou, and J. Zhao, "Relation classification via convolutional deep neural network," in *Proc. COLING*, 2014, pp. 2335–2344.

[31] X. Liu, L. Mou, H. Cui, Z. Lu, and S. Song, "Finding decision jumps in text classification," *Neurocomputing*, vol. 371, pp. 177–187, Jan. 2020.

[32] L. Cai, Y. Song, T. Liu, and K. Zhang, "A hybrid BERT model that incorporates label semantics via adjustive attention for multi-label text classification," *IEEE Access*, vol. 8, pp. 152183–152192, 2020.

[33] A. Mueller et al., "Label semantic aware pre-training for few-shot text classification," 2022, *arXiv:2204.07128*.

[34] M. Liu, L. Liu, J. Cao, and Q. Du, "Co-attention network with label embedding for text classification," *Neurocomputing*, vol. 471, pp. 61–69, Jan. 2022.

[35] X. Zhu, Z. Peng, J. Guo, and S. Dietze, "Generating effective label description for label-aware sentiment classification," *Expert Syst. Appl.*, vol. 213, Mar. 2023, Art. no. 119194.

[36] C. Du, Z. Chen, F. Feng, L. Zhu, T. Gan, and L. Nie, "Explicit interaction model towards text classification," in *Proc. AAAI*, vol. 33, 2019, pp. 6359–6366.

[37] K. Rivas Rojas, G. Bustamante, A. Oncevay, and M. A. S. Cabezudo, "Efficient strategies for hierarchical text classification: External knowledge and auxiliary tasks," in *Proc. ACL*, 2020, pp. 2252–2257.

[38] H. Wu, S. Qin, R. Nie, J. Cao, and S. Gorbachev, "Effective collaborative representation learning for multilabel text categorization," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 10, pp. 5200–5214, Oct. 2022.

[39] X. Wang, L. Zhao, B. Liu, T. Chen, F. Zhang, and D. Wang, "Concept-based label embedding via dynamic routing for hierarchical text classification," in *Proc. ACL-IJCNLP*, 2021, pp. 5010–5019.

[40] N. Wang, W. Zhou, and H. Li, "Contrastive transformation for self-supervised correspondence learning," in *Proc. AAAI*, 2021, pp. 10174–10182.

[41] Z. Wu, S. Wang, J. Gu, M. Khabsa, F. Sun, and H. Ma, "CLEAR: Contrastive learning for sentence representation," 2020, *arXiv:2012.15466*.

[42] P. Khosla et al., "Supervised contrastive learning," in *Proc. NeurIPS*, 2020, pp. 1–13.

[43] T. T. Cai, J. Frankle, D. J. Schwab, and A. S. Morcos, "Are all negatives created equal in contrastive instance discrimination?" 2020, *arXiv:2010.06682*.

[44] T. Chen, S. Kornblith, M. Norouzi, and G. E. Hinton, "A simple framework for contrastive learning of visual representations," 2020, *arXiv:2002.05709*.

[45] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," 2019, *arXiv:1911.05722*.

[46] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, "Masked autoencoders are scalable vision learners," 2021, *arXiv:2111.06377*.

[47] T. Gao, X. Yao, and D. Chen, "SimCSE: Simple contrastive learning of sentence embeddings," in *Proc. EMNLP*, 2021, pp. 6894–6910.

[48] Y. Li, P. Hu, Z. Liu, D. Peng, J. T. Zhou, and X. Peng, "Contrastive clustering," in *Proc. AAAI*, 2021, pp. 8547–8555.

[49] M. Yang, Y. Li, Z. Huang, Z. Liu, P. Hu, and X. Peng, "Partially view-aligned representation learning with noise-robust contrastive loss," in *Proc. CVPR*, Jun. 2021, pp. 1134–1143.

[50] M. Yang, Y. Li, P. Hu, J. Bai, J. Lv, and X. Peng, "Robust multi-view clustering with incomplete information," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 1, pp. 1055–1069, Jan. 2023.

[51] M. E. Peters et al., "Deep contextualized word representations," 2018, *arXiv:1802.05365*.

[52] R. Sennrich, B. Haddow, and A. Birch, "Neural machine translation of rare words with subword units," in *Proc. ACL*, 2016, pp. 1715–1725.

[53] Q. Chen, X. Zhu, Z.-H. Ling, S. Wei, H. Jiang, and D. Inkpen, "Enhanced LSTM for natural language inference," in *Proc. ACL*, 2017, pp. 1657–1668.

[54] K. Zhang, E. Chen, Q. Liu, C. Liu, and G. Lv, "A context-enriched neural network method for recognizing lexical entailment," in *Proc. AAAI*, 2017, pp. 3127–3133.

[55] L. Mou et al., "Natural language inference by tree-based convolution and heuristic matching," in *Proc. ACL*, 2016, pp. 130–136.

[56] J. Weeds, D. Clarke, J. Reffin, D. Weir, and B. Keller, "Learning to distinguish hypernyms and co-hyponyms," in *Proc. COLING*, 2014, pp. 2249–2259.

[57] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," in *Proc. CVPR*, Jun. 2015, pp. 815–823.

[58] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *Proc. ICML*, 2020, pp. 1597–1607.

[59] Y. Tay, A. T. Luu, and S. C. Hui, "Co-stack residual affinity networks with multi-level attention refinement for matching text sequences," in *Proc. ACL*, 2018, pp. 4492–4502.

[60] R. Yang, J. Zhang, X. Gao, F. Ji, and H. Chen, "Simple and effective text matching with richer alignment features," in *Proc. ACL*, 2019, pp. 4699–4709.

[61] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, "Albert: A lite BERT for self-supervised learning of language representations," in *Proc. ICLR*, 2020, pp. 1–13.

[62] G. Wang et al., "Joint embedding of words and labels for text classification," in *Proc. ACL*, 2018, pp. 2321–2331.

[63] M. Marelli, L. Bentivogli, M. Baroni, R. Bernardi, S. Menini, and R. Zamparelli, "SemEval-2014 task 1: Evaluation of compositional distributional semantic models on full sentences through semantic relatedness and textual entailment," in *Proc. SemEval*, 2014, pp. 1–8.

[64] T. Khot, A. Sabharwal, and P. Clark, "SciTaiL: A textual entailment dataset from science question answering," in *Proc. AAAI*, 2018, pp. 5189–5197.

[65] J. Howard and S. Ruder, "Universal language model fine-tuning for text classification," in *Proc. ACL*, 2018, pp. 328–339.

[66] W. Lan and W. Xu, "Neural network models for paraphrase identification, semantic textual similarity, natural language inference, and question answering," in *Proc. COLING*, 2018, pp. 3890–3902.

[67] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, Nov. 2008.

Kun Zhang (Member, IEEE) received the Ph.D. degree in computer science and technology from the University of Science and Technology of China, Hefei, China, in 2019.

He is currently a Faculty Member with the Hefei University of Technology (HFUT), Hefei. He has published several articles in refereed journals and conferences, such as the IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS: SYSTEMS, IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, *ACM Transactions on Knowledge Discovery from Data*, Association for the Advancement of Artificial Intelligence (AAAI), ACM Sigkdd Conference on Knowledge Discovery and Data Mining (KDD), Annual Meeting of the Association for Computational Linguistics (ACL), and The IEEE International Conference on Data Mining (ICDM). His research interests include natural language understanding and recommendation systems.

Dr. Zhang was a recipient of the KDD 2018 Best Student Paper Award.

Le Wu (Member, IEEE) received the Ph.D. degree in computer science from the University of Science and Technology of China (USTC), Hefei, China, in 2015.

She is currently a Professor with the Hefei University of Technology (HFUT), Hefei. She has published several papers in refereed journals and conferences, such as the IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, the ACM Transactions on Intelligent Systems and Technology, Association for the Advancement of Artificial Intelligence (AAAI), International Joint Conference on Artificial Intelligence (IJCAI), ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD), SIAM International Conference on Data Mining (SDM), and The IEEE International Conference on Data Mining (ICDM). Her general areas of research are data mining, recommender systems, and social network analysis.

Dr. Wu was a recipient of the Best of SDM 2015 Award.

Guangyi Lv received the Ph.D. degree in computer science and technology from the University of Science and Technology of China, Hefei, China, in 2019.

He is currently an Advisory Researcher at Lenovo Research, Beijing, China. His major research interests include deep learning, natural language processing, and adaptive artificial intelligence (AI). He has published several papers in refereed journals and conferences, such as the IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS: SYSTEMS, the IEEE TRANSACTIONS ON BIG DATA, Association for the Advancement of Artificial Intelligence (AAAI), International Joint Conference on Artificial Intelligence (IJCAI), The IEEE International Conference on Data Mining (ICDM). His major research interests include deep learning, natural language processing, and recommendation systems.

Enhong Chen (Senior Member, IEEE) received the Ph.D. degree in computer science from the University of Science and Technology of China (USTC), Hefei, China, in 1996.

He is currently a Professor and the Vice Dean of the School of Computer Science, USTC. He has authored or coauthored more than 100 papers in refereed conferences and journals, including the IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, the IEEE TRANSACTIONS ON MOBILE COMPUTING, ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD), The IEEE International Conference on Data Mining (ICDM), NIPS, and CIKM. His research is supported by the National Science Foundation for Distinguished Young Scholars of China. His general areas of research include data mining and machine learning, social network analysis, and recommender systems.

Dr. Chen was on the Program Committee of numerous conferences, including KDD, ICDM, and SIAM International Conference on Data Mining (SDM).

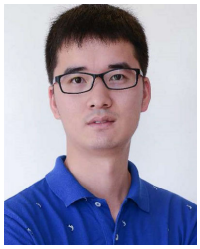
Shulan Ruan received the B.S. degree from Hunan University, Changsha, China, in 2018. He is currently pursuing the Ph.D. degree with the School of Computer Science and Technology, University of Science and Technology of China, Hefei, China.

He has published several papers in Association for the Advancement of Artificial Intelligence (AAAI) and ICME. His research interests include sentiment analysis, computer vision, and natural language processing.

Jing Liu (Member, IEEE) received the B.E. and M.S. degrees from Shandong University, Jinan, Shandong, China, in 2001 and 2004, respectively, and the Ph.D. degree from the Institute of Automation, Chinese Academy of Sciences, Beijing, in 2008.

She is currently a Professor with the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences. Her current research interests include deep learning, image content analysis and classification, multimedia, understanding, and retrieval.

Zhiqiang Zhang is currently a Staff Engineer at Ant Group Company Ltd., Hangzhou, China. He has led a team to build an industrial graph machine learning system, AGL, Ant Group. He has published more than 30 papers in top-tier machine learning and data mining conferences, including The Conference on Neural Information Processing Systems (NeurIPS), International Conference on Very Large Data Bases (VLDB), ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD), and Association for the Advancement of Artificial Intelligence (AAAI). His research interests mainly focus on graph machine learning.



Jun Zhou is currently a Senior Staff Engineer at Ant Group Company Ltd., Hangzhou, China. He has participated in the development of several distributed systems and machine learning platforms in Alibaba and Ant Group, such as Apsaras (Distributed Operating System), MaxCompute (Big Data Platform), and KunPeng (Parameter Server). He has published more than 40 papers in top-tier machine learning and data mining conferences, including International Conference on Very Large Data Bases (VLDB), International World Wide Web Conference (WWW), The Conference on Neural Information Processing Systems (NeurIPS), and Association for the Advancement of Artificial Intelligence (AAAI). His research mainly focuses on machine learning and data mining.

Meng Wang (Fellow, IEEE) received the B.E. and Ph.D. degrees from the University of Science and Technology of China (USTC), Hefei, China, in 2003 and 2008, respectively.

He is currently a Professor with the Hefei University of Technology (HFUT), Hefei. His current research interests include multimedia content analysis, computer vision, and pattern recognition. He has authored or coauthored more than 200 book chapters, journal articles, and conference papers in these areas.

Dr. Wang was a recipient of the ACM SIGMM Rising Star Award 2014. He is an Associate Editor of the IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, and the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS.