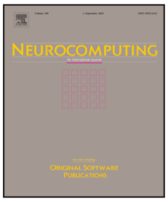
Contents lists available at ScienceDirect

Neurocomputing

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



CRMSP: A semi-supervised approach for key information extraction with Class-Rebalancing and Merged Semantic Pseudo-Labeling

Qi Zhang, Yonghong Song *, Pengcheng Guo, Yangyang Hui

School of Software Engineering, Xi'an Jiaotong University, No. 28, Xianning West Road, Xi'an City, 710049, Shaanxi Province, China

ARTICLE INFO

Communicated by Y. Yan

Keywords:
Semi-supervised learning
Key information extraction
Long-tailed distribution
Semantic Pseudo-Labeling

ABSTRACT

There is a growing demand in the field of Key Information Extraction (KIE) to apply semi-supervised learning (SSL) to save manpower and costs, as training document data using fully-supervised methods requires labor-intensive manual annotation. The main challenges of applying SSL in the KIE are (1) underestimation of the confidence of tail classes in the long-tailed distribution and (2) difficulty in achieving intra-class compactness and inter-class separability of tail features. To address these challenges, we propose a novel semi-supervised approach for KIE with Class-Rebalancing and Merged Semantic Pseudo-Labeling (CRMSP). Firstly, the Class-Rebalancing Pseudo-Labeling (CRP) module introduces a reweighting factor to rebalance pseudo-labels, increasing attention to tail classes. Secondly, we propose the Merged Semantic Pseudo-Labeling (MSP) module to cluster tail features of unlabeled data by assigning samples to Merged Prototypes (MP). Additionally, we designed a new contrastive loss specifically for MSP. Extensive experimental results on three well-known benchmarks demonstrate that CRMSP achieves state-of-the-art performance. Remarkably, CRMSP achieves 3.24% f1-score improvement over state-of-the-art on the CORD.

1. Introduction

Key Information Extraction (KIE) as the downstream task of Optical Character Recognition (OCR) is the process of extracting structured information from documents. KIE generally includes tasks such as named entity recognition and relation extraction, structured information extraction, and document classification. KIE has various applications in real-life scenarios, including bill processing, medical record handling, contract analysis, and resume processing. KIE is a challenging task since documents involve different types of information, including images, text, and layout. Recently, many multimodal pre-trained methods [1–3] for KIE have been proposed to fickle this problem. However, these multimodal pre-trained methods require annotation for multiple types of information, which further increases time and manpower costs.

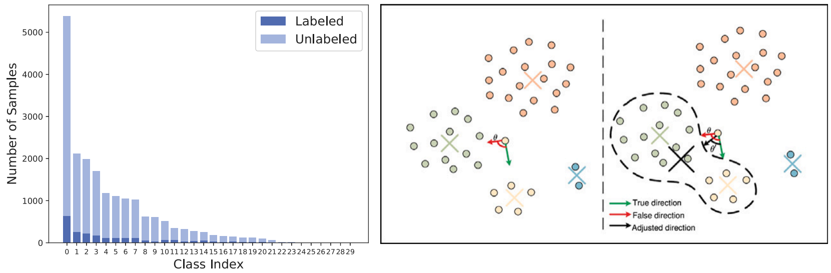
Semi-supervised learning (SSL) [4] tackles situations with limited labeled and abundant unlabeled data [5–9], bridging the gap between supervised and unsupervised learning for enhanced model performance. Existing SSL approaches [5,10] are to perform consistency regularization between weakly and strongly augmented views of unlabeled data based on the pseudo-labels predicted by the model as targets, thereby mitigating the model's sensitivity to small variations in similar samples within the input space. The performance of these SSL methods based on consistency regularization depends on whether the classes are balanced and the intra-class compactness and inter-class separability of the model in the feature space.

Specifically, the first one is that the confidence of tail classes in the long-tailed distribution [11] is underestimated, leading to the model exhibiting higher confidence in predicting samples from the head classes. As shown in Fig. 1(a), both labeled and unlabeled data exhibit a long-tailed distribution. This phenomenon implies that pseudo-labels are more likely to belong to head classes with higher probabilities and less likely to belong to tail classes with lower probabilities. As the number of iterations increases, this imbalance in the long-tailed distribution tends to worsen.

Secondly, it is hard to achieve intra-class compactness and inter-class separability of tail classes in unlabeled feature space [12]. To help the model learn richer representations, prototypes [13,14] are commonly introduced into the model. Each prototype can be seen as the representative features of a specific class of samples. The model calculates semantic pseudo-labels based on these prototypes. Following this, consistency regularization is applied to learn and enhance features. As shown in Fig. 1(b), when classifying yellow unlabeled samples of tail classes, the sample points are closer to the "green" prototype than to "yellow". This results in "yellow" sample points being pushed away from the true direction and towards the wrong direction in the feature

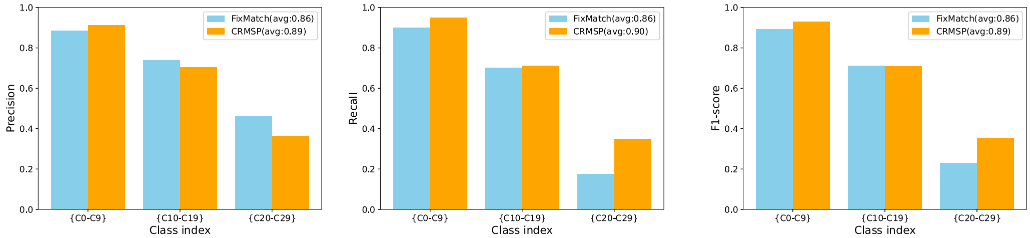
* Corresponding author.
E-mail addresses: qi_zhang@stu.xjtu.edu.cn (Q. Zhang), songyh@xjtu.edu.cn (Y. Song), bedlexmunaxl@stu.xjtu.edu.cn (P. Guo), YangyangHui@stu.xjtu.edu.cn (Y. Hui).

https://doi.org/10.1016/j.neucom.2024.128907
Received 19 May 2024; Received in revised form 16 October 2024; Accepted 7 November 2024
Available online 16 November 2024



(a) Class distribution (b) Dilemma of tail classes

Fig. 1. Analysis on class distribution, and the dilemma of tail classes.



(a) Precision of PL (b) Recall of PL (c) F1-score of PL

Fig. 2. Comparison of precision, recall and f1-score of pseudo-labels generated by FixMatch and CRMSP. "PL" represents Pseudo-Labels. The horizontal axis {C0–C9}, {C10–C19} and {C20–C29} represents the collections of head, medium, and tail classes, respectively. The vertical axis shows the average precision, recall and f1-score of all classes in these three class collections for each column, and "avg." indicates the average precision, recall and f1-score of the {C0–C29} classes.

space. Therefore, the existence of the deviation angle θ causes semantic pseudo-labels to be biased to head classes over the tail classes, and this imbalance will increase as training progresses, leading to a degradation in model performance.

To address these challenges, we propose a semi-supervised approach for KIE with Class-Rebalancing and Merged Semantic Pseudo-Labeling (CRMSP). Firstly, to augment the model's attention towards tail classes, we introduce the Class-Rebalancing Pseudo-Labeling (CRP) module that enhances the weight of pseudo-labels of tail classes and reduces the weight of pseudo-labels of head classes with a reweighting factor. It improves the recall of tail classes compared to the classical SSL method FixMatch [5] while maintaining a high precision of head classes, resulting in a higher f1-score, as shown in Fig. 2.

Secondly, to enhance the intra-class compactness and separability from other classes of tail classes, we propose the Merged Semantic Pseudo-Labeling (MSP) module. This module sorts the generated pseudo-labels in descending order, identifies the Top-K classes, and aggregates the features of these K classes in the memory bank into a super-class. The merged prototype (MP) is then calculated. By utilizing the clustering of merged prototypes, the semantic pseudo-labels generated in this way push the features of tail samples closer to the prototype of the super-class, rather than pushing the features closer to prototypes of head classes.

Additionally, we designed a new contrastive loss specifically for the merged semantic pseudo-labels, whose effectiveness is demonstrated in Table 4. Extensive experimental results indicate that the MSP module improves the performance of tail classes.

To the best of our knowledge, CRMSP is the first semi-supervised learning method in the field of KIE. Based on a multi-modal model, CRMSP has designed a semi-supervised learning method that fully utilizes text, image, and layout information, which is different from previous semi-supervised methods that only utilize image or text information from CV or NLP models. The main contributions are summarized as follows:

- We propose a semi-supervised approach for KIE with Class-Rebalancing and Merged Semantic Pseudo-Labeling (CRMSP), utilizing a large number of unlabeled documents, significantly reducing the annotation costs, and improving the generalizability of the model.
- To solve the problem of underestimation of the confidence of tail classes in the long-tailed distribution, we proposed the Class-Rebalancing Pseudo-Labeling (CRP) module.
- We propose the Merged Semantic Pseudo-Labeling (MSP) module to fickle the difficulty in achieving intra-class compactness and inter-class separability of tail classes in unlabeled feature space.

2. Related work

2.1. Key information extraction

Transformer-based pre-training has demonstrated success across various KIE tasks, where extensive unlabeled document datasets are leveraged for model pre-training, preceding fine-tuning on downstream tasks. Numerous existing frameworks [1–3,15] have investigated pre-training approaches on documents. LayoutLM [15] achieved significant improvements in various document understanding tasks by jointly pre-training text and layout. LayoutLMv2 [1] greatly enhanced the model's image understanding capability by integrating visual feature information into the pre-training process. LayoutLMv3 [2] overcame the differences between text and image in pre-training objectives and promoted multi-modal representation learning. Our approach utilizes these multi-modal models based on image, text, and layout as encoders, extending the scope of semi-supervised methods to the KIE domain.

2.2. Semi-supervised learning

SSL is a learning approach focused on building models that leverage both labeled and unlabeled data. While unlabeled data is crucial for SSL, generating pseudo-labels from model predictions remains a challenge. Existing approaches, including pseudo-labeling [16], consistency regularization [17,18], generative methods [19,20] and hybrid methods [5–9,21]. However, pseudo-labels can introduce bias, particularly in the presence of imbalanced data, adversely affecting model performance. To mitigate this issue, previous works have explored various strategies such as threshold adjustment [6,7,9], incorporating additional classifiers [14,22]. However, designing dynamic thresholds is complex and computationally intensive. In our work, we directly incorporate an additional branch for semantic pseudo-label classification, which effectively promotes intra-class compactness and inter-class separability for imbalanced classes, without the need for designing complex dynamic threshold strategies.

2.3. Imbalanced learning

Imbalanced fully-supervised learning and semi-supervised learning are of great interest in theory. Imbalanced fully-supervised learning mainly includes two types of techniques: resampling and data augmentation. [23] is a typical resampling method, which uses the interpolation between a given minority sample and its nearest minority neighbors to create new samples. [24] proposed a dual sampling strategy to balance the performance between head classes and tail classes and to avoid overfitting. Mixup-based methods are effective data augmentation methods, aiming to enrich the effective semantic information for tail classes. For example, [25] proposed an adaptive image-mixing method, which is able to generate semantically reasonable and meaningful mixed images for tail classes. [26] proposed augmenting tail classes by grafting diverse semantic information from head classes, referred to as head-to-tail fusion (H2T). [27] proposed a novel knowledge-transferring-based calibration method by estimating the importance weights of tail classes to transfer knowledge from head classes to obtain the target probability density of tail classes. Recent works in the field of imbalanced semi-supervised learning include resampling [23,28] and reweighting [29] which rebalance the contribution of each class, while others focus on reweighting the given loss function by a factor inversely proportional to the sampling frequency in a class-wise manner. [29] proposed a suppressed consistency loss to suppress the loss on minority classes. [30] proposed Distribution Aligning Refinery (DARP) to refine pseudo-labels for SSL under assuming class-imbalanced training distributions. CReST proposed a re-sampling method to iteratively refine the model by supplementing the labeled set with high-quality pseudo-labels, where minority classes are updated more aggressively than majority classes. DASO adaptively blends the linear and semantic pseudo-labels within each class to mitigate the overall bias across the class for imbalanced semi-supervised learning. In our work, we alleviate the class-imbalanced problem by directly rebalancing pseudo-labels according to distributions between head and tail classes instead of designing complicated reweighting losses.

3. Proposed method

3.1. Preliminaries

For a C-class semi-supervised classification problem, let $\mathcal{X} = \{(x_b, y_b)\}_{b=1}^{B}$ be a batch of B labeled samples, where x_b are the training samples and y_b are the ground-truth, $y_b \in \mathcal{Y} = \{1, \dots, C\}$. Meanwhile, let $\mathcal{U} = \{u_b\}_{b=1}^{\mu B}$ be a batch of μB unlabeled samples, where the hyperparameter μ is used to control the batch size of unlabeled samples. Note that the underlying ground truth $\hat{y}$ of unlabeled data may be different from labeled data, $\hat{y} \in \mathcal{Y}$, $\mathcal{Y} = \{1, \dots, C\}$.

For the labeled data, the input x_b is paired with the label y_b to train the base model $f(\cdot)$ through calculating supervised loss $\mathcal{L}_{sup}$, generating features z_b. For the unlabeled data, unlabeled samples are sent to the base model $f(\cdot)$ as inputs after weak augmentation $\mathcal{A}_w$ and strong augmentation $\mathcal{A}_s$. Both are followed by a classification head $h(\cdot)$ and a projection head $g(\cdot)$ to get $p^w = h \circ f(\mathcal{A}_w(u))$, $z^w = g \circ f(\mathcal{A}_w(u))$, $p^s = h \circ f(\mathcal{A}_s(u))$ and $z^s = g \circ f(\mathcal{A}_s(u))$. The Class-Rebalancing Pseudo-labeling module is employed to alleviate the imbalance problem of pseudo-labels. The rebalanced pseudo-labels $\hat{p} \in \mathbb{R}^C$ are then assigned to calculate the unsupervised loss $\mathcal{L}_{un}$. The Merged Semantic Pseudo-Labeling module generates merged semantic pseudo-labels of unlabeled features with the Merged Prototypes $\hat{\mathbf{C}}$, which is used to compute the contrastive loss $\mathcal{L}_{ctr}$. The overall framework is shown in Fig. 3.

3.2. Class-rebalancing pseudo-labeling

Due to the smaller sample size in the tail classes compared to the head classes, the model tends to generate lower confidence when predicting tail data. The approach in FixMatch [5], which filters out samples based on a fixed threshold applied to the highest confidence, overlooks the numerical disadvantage of the tail data. Experiments show that the predictive distribution of labeled samples is generally positively correlated with the distribution of unlabeled samples. We estimate the approximate sample distribution by calculating the exponential moving average (EMA) of the model's confidence predictions for labeled data p, which we call $\tilde{p}$. Note that EMA preserves previous information through a weighted average, smoothing the process of updating data. At the tth iteration, we compute $\tilde{p}_t$ as:

$$\tilde{p}_t = \begin{cases} \frac{1}{C}, & \text{if } t = 0, \\ \lambda \tilde{p}_{t-1} + (1-\lambda)\frac{1}{B}\sum_{b=1}^{B} p, & \text{otherwise,} \end{cases} \tag{1}$$

where λ is a smoothing factor.

3.2.1. Reweighting pseudo-labels

We observed that the pseudo-labels produced by the model are biased towards the head classes. To augment the model's attention towards tail classes, we introduced a reweighting factor β that enhances the weight of tail classes while correspondingly reducing that of head classes.

According to Bayes' theorem:

$$p(y|x) = \frac{p(x|y) \cdot p(y)}{p(x)} \tag{2}$$

which implies $p(y|x) \propto p(y) \cdot p(x|y)$. When our model generates the predicted score $p(y|x)$, we are essentially updating our confidence in the label y using the prior knowledge $p(y)$ and the labeled data $p(x|y)$. When updating with $p(y|x)$, the prior distribution of labels $p(y)$ is highly skewed towards head and tail classes, which results in many tail labels having very low probabilities. To address this issue, we propose reweighting the predicted scores as a post-processing step. Specifically, the model's confidence predictions p_i^w for weakly-augmented data is multiplied by the reweighting factor β_i to get more balanced confidence predictions p_i':

$$p_i' = Normalize(\beta_i \times p_i^w), 1 \le i \le C \tag{3}$$

The rebalanced pseudo-labels $\hat{p}_b$ are generated by $argmax(\cdot)$ in a batch:

$$\hat{p}_b = argmax(p_b'). \tag{4}$$

By rebalancing the pseudo-labels, tail labels are more likely to be chosen when filtering pseudo-labels with a fixed threshold for prediction.

3.2.2. Reweighting mapping function

Given the update rule Eq. (1) of the model's EMA confidence prediction $\tilde{p}_t$, we aim to prove that $\tilde{p}_t \propto p(y|x)$. When $t = 0$, the initial value is $\tilde{p}_0 = \frac{1}{K}$. For $t > 0$, the iterative relationship is:

$$\tilde{p}_t = \lambda \tilde{p}_{t-1} + (1-\lambda)\frac{1}{B}\sum_{b=1}^{B} p(y \mid x) \tag{5}$$

To understand the relationship between $\lambda \tilde{p}_t$ and $p(y|x)$, we can expand $\tilde{p}_t$ step by step:

$$\tilde{p}_t = \lambda(\lambda \tilde{p}_{t-2} + (1-\lambda)\frac{1}{B}\sum_{b=1}^{B} p(y|x)) + (1-\lambda)\frac{1}{B}\sum_{b=1}^{B} p(y|x). \tag{6}$$

Expanding further:

$$\tilde{p}_t = \lambda^2 \tilde{p}_{t-2} + \lambda(1-\lambda)\frac{1}{B}\sum_{b=1}^{B} p(y|x) + (1-\lambda)\frac{1}{B}\sum_{b=1}^{B} p(y|x). \tag{7}$$

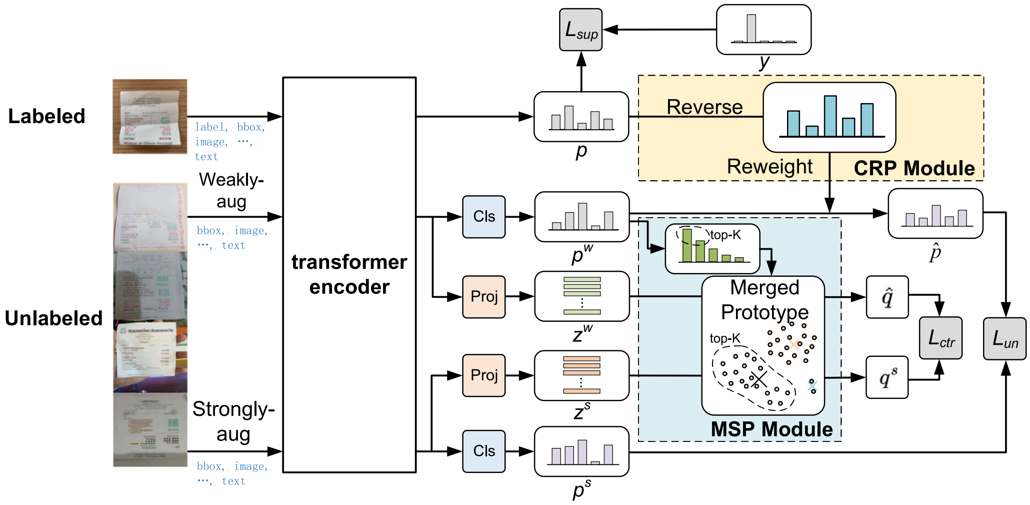


Fig. 3. Framework of the proposed Class-Rebalancing and Merged Semantic Pseudo-Labeling (CRMSP). Labeled and unlabeled samples are from the training data mini-batch.

Continuing to expand:

$$\tilde{p}_t = \lambda^t \tilde{p}_0 + \sum_{k=0}^{t-1} \lambda^k (1-\lambda) \frac{1}{B} \sum_{b=1}^{B} p(y|x). \tag{8}$$

From the expanded iterative formula, we can see that $\tilde{p}_t$ is determined by the initial value $\tilde{p}_0 = \frac{1}{K}$ and the weighted average of past $p(y|x)$ values.

As t increases, $\lambda^t \tilde{p}_0$ will gradually approach zero since $0 < \lambda < 1$. Thus, for large t, $\tilde{p}_t$ will be primarily determined by the second term, which is:

$$\tilde{p}_t \approx \sum_{k=0}^{t-1} \lambda^k (1-\lambda) \frac{1}{B} \sum_{b=1}^{B} p(y|x). \tag{9}$$

From the iterative formula, it is evident that $\tilde{p}_t$ is composed of the weighted average of $p(y|x)$. Since $\tilde{p}_t$ is updated iteratively, if $\tilde{p}_{t-1}$ is close to $p(y|x)$, then $\tilde{p}_t$ will maintain a proportional relationship with $p(y|x)$ at each iteration. Thus, it is proven that $\tilde{p}_t \propto p(y|x)$.

Based on the proportional relationship $p(y|x) \propto p(y) \cdot p(x|y)$ derived in Eq. (2), we can conclude $\tilde{p}_t \propto p(y) \cdot p(x|y)$. During the training process for long-tailed data, although $p(y)$ is unknown, it can be estimated by calculating $\tilde{p}_t$.

We first perform the reverse operation on $\tilde{p}_t$ to obtain the reweighting factor β:

$$\beta_i = \mathcal{M}(\tilde{p}_t^{(i)}), 1 \le i \le C \tag{10}$$

$\mathcal{M}(\cdot)$ is a monotonically decreasing mapping function (a minimum value not less than 0) that ensures a higher weight is assigned to classes with a smaller proportion in the predicted distribution, while classes with a larger proportion in the predicted distribution receive a lower weight. e.g., $\mathcal{M}(x) = 1 - x/T$, where T is a temperature hyperparameter, $Normalize(\cdot)$ is the normalized operation defined as $x_i' = x_i / \sum_{j=1}^{n} x_j, i \in (1, \dots, n)$. We have presented the ablation experiment results for three mapping functions in Section 4.4.7.

Through the above calculation, we have implicitly used a balanced labeling probability function $p(y|x) = \frac{1}{\beta} \cdot p(x|y)$. In this way, we obtain more balanced prior knowledge from the labeled data, which, by applying Eq. (3), is used to achieve more balanced pseudo-labels for the unlabeled data.

3.3. Merged semantic pseudo-labeling

To obtain semantic pseudo-labels from a feature perspective, DASO [14] involves prototype clustering, which updates the dynamic memory bank with features and ground-truths of labeled data. However, due to the significantly smaller number of tail samples compared to head classes, the tail features in this memory bank are inherently limited. Consequently, the computed tail prototypes lack representation, and it is inappropriate to assume that all tail features are concentrated around this prototype. The semantic pseudo-labels are computed by merely comparing tail samples to this unrepresentative prototype push tail features close to prototypes of other classes, which is detrimental to achieving intra-class compactness and inter-class separability of tail classes.

3.3.1. EMA model

The basic assumption in SSL is the smoothness assumption [31,32]: if two data points are close in high-density regions, their corresponding outputs should also be close. Mean Teacher [17] utilizes this assumption by using unlabeled data. In practice, augmented samples are generated by adding small perturbations to the original samples, and they should have consistent predictions in both Teacher and Student models, achieved through consistency constraints. The Teacher model is essentially an EMA model of the Student. The EMA model provides guidance for updating the parameters of the base model. Therefore, their weights are tightly coupled. The parameter (θ') of the EMA model is updated with a weighted average of the current parameter (θ) of the base model:

$$\theta_t' = \alpha \cdot \theta_{t-1}' + (1-\alpha) \cdot \theta \tag{11}$$

where α is a smoothing factor ($0 < \alpha < 1$).

3.3.2. Merged Prototypes (MP) generation

We first build a set of basic prototypes $\mathbf{C} = \{c_i\}_{i=1}^{C}$ from $\mathcal{X}$. The basic prototype c_i for every class is efficiently calculated by averaging the feature representations in the dynamic memory bank $\mathbf{Q} = \{Q_i\}_{i=1}^{C}$, $Q_i = \{z_j\}_{j=1}^{maxsize}$, where Q_i is a queue with a max size. We update $\mathbf{Q}$ every iteration by pushing new features z_b and labels y_b from a batch of labeled data.

Then we determine to construct the super-class for each batch. Based on a common understanding: for an unlabeled sample, if the confidences of several classes are close, their corresponding feature representations in the feature space are close. In such cases, we merge these top-K proximate classes. We achieve this by sorting the confidence predictions p_b^w obtained from the weak augmentation branch in descending order, resulting in the sorted confidence predictions s_b^w:

$$s_b^w = \mathrm{sort}(p_b^w, \text{descending}) = \{p_{\sigma(1)}^w, \dots, p_{\sigma(K)}^w, \dots, p_{\sigma(C)}^w\} \tag{12}$$

$$\{\sigma(1), \sigma(2), \dots, \sigma(C)\} = \arg\ \mathrm{sort}(s_b^w) \tag{13}$$

where $\{\sigma(1), \dots, \sigma(C)\}$ represents the order in which classes are arranged in descending order.

After obtaining this order, we merge the features corresponding to the top-K classes $Q_{\sigma(K)}$ in the dynamic memory bank $\mathbf{C}$ to get the new dynamic memory bank is $\hat{\mathbf{Q}} = \{\hat{Q}_i\}_{i=1}^N$, where $N = C - K + 1$:

$$\hat{Q}_i = \begin{cases} Q_{\sigma(1)} \cup Q_{\sigma(2)} \cup \cdots \cup Q_{\sigma(K)}, & \text{if } i = 1 \\ Q_{\sigma(K+i-1)}, & \text{otherwise} \end{cases} \tag{14}$$

For Merged prototypes $\hat{\mathbf{C}} = \{\hat{c}_i\}_{i=1}^N$, each $\hat{c}_i$ is computed by taking the average of the features in the queue $\hat{Q}_i$. Note that $\hat{\mathbf{Q}} = \{\hat{Q}_i\}_{i=1}^N$, where $N = C - K + 1$, needs to be regenerated in each batch based on the guidance of Top-K from the unique $\mathbf{Q} = \{Q_i\}_{i=1}^C$. This means that $\mathbf{Q}$ remains the same across all batches, whereas $\hat{\mathbf{Q}}$ is different in each batch.

3.3.3. Semantic pseudo-labels

In a batch, the merged semantic predictions q_b^w and q_b^s of the super class space is computed from z_b^w, z_b^s with the merged prototype $\hat{\mathbf{C}}$:

$$q_b^w = Sim(z_b^w, \hat{\mathbf{C}}) = \frac{\langle z_b^w, \hat{\mathbf{C}} \rangle}{\|z_b^w\| \|\hat{\mathbf{C}}\|} / T_{proto} \tag{15}$$

$$q_b^s = Sim(z_b^s, \hat{\mathbf{C}}) = \frac{\langle z_b^s, \hat{\mathbf{C}} \rangle}{\|z_b^s\| \|\hat{\mathbf{C}}\|} / T_{proto} \tag{16}$$

where T_{proto} is a temperature hyperparameter, and $Sim(\cdot)$ denotes cosine similarity, $\langle \cdot, \cdot \rangle$ represents the dot product operation, and $\| \cdot \|$ represents the L2 norm.

The merged semantic pseudo-labels $\hat{q}_b$ are generated by $argmax(\cdot)$ in a batch:

$$\hat{q}_b = argmax(q_b^w). \tag{17}$$

3.4. Loss function

Following the SSL paradigm, the first two items are supervised loss $\mathcal{L}_{sup}$ and unsupervised loss $\mathcal{L}_{un}$, respectively. In addition, we include a contrastive loss $\mathcal{L}_{ctr}$ to compute the distance of two semantic similarities. The loss minimized by CRMSP is simply:

$$\mathcal{L}_{total} = \mathcal{L}_{sup} + \lambda_{un}\mathcal{L}_{un} + \lambda_{ctr}\mathcal{L}_{ctr}, \tag{18}$$

$$\mathcal{L}_{sup} = \frac{1}{B} \sum_{b=1}^{B} \mathcal{H}(p_b, y_b), \tag{19}$$

$$\mathcal{L}_{un} = \frac{1}{\mu B} \sum_{b=1}^{\mu B} \mathbb{1}(max(p_b') \geq \tau) \mathcal{H}(\hat{p}_b, p_b^s), \tag{20}$$

$$\mathcal{L}_{ctr} = \frac{1}{\mu B} \sum_{b=1}^{\mu B} \mathcal{H}(\hat{q}_b, q_b^s). \tag{21}$$

where λ_{un} and λ_{ctr} are fixed scalar hyperparameters denoting the relative weight of the unsupervised loss and contrastive loss, and $\mathbb{1}$ is all-one vector, τ represents a fixed threshold. $\mathcal{H}$ represents cross-entropy.

3.5. Pseudo code

Algorithm 1 presents the algorithm of the entire CRMSP during the training phase.

```
Data: A batch of labeled and unlabeled samples
      $\mathcal{X} = \{(x_b, y_b)\}_{b=1}^B$ and $\mathcal{U} = \{u_b\}_{b=1}^{\mu B}$, base model $f(\cdot)$,
      classification head and projection head: $h(\cdot)$ and $g(\cdot)$
Input  : $\lambda_{un}$: weight of the unsupervised loss, $\lambda_{ctr}$: weight of
         the contrastive loss, $\tau$: a fixed threshold
Output: Trained model parameters

for epoch in range(num_epochs) do
    for batch in unlabeled_data_batches do
        {Supervised Learning with labeled data};
        $p_b = h \circ f(x_b)$,
        $\tilde{p}_t = \begin{cases} \frac{1}{K}, & \text{if } t = 0, \\ \lambda \tilde{p}_{t-1} + (1-\lambda)\frac{1}{B}\sum_{b=1}^{B} p_b, & \text{otherwise,} \end{cases}$;
        $\mathcal{L}_{sup} = \frac{1}{B} \sum_{b=1}^{B} \mathcal{H}(p_b, y_b)$;
        {Unsupervised Learning with unlabeled data};
        $p_b^w = h \circ f(\mathcal{A}_w(u_b))$, $p_b^s = h \circ f(\mathcal{A}_s(u_b))$;
        $\beta = Normalize(\mathcal{M}(\tilde{p}_t))$, $p_b' = Normalize(p_b^w \times \beta)$;
        $\hat{p}_b = argmax(p_b')$;
        $\mathcal{L}_{un} = \frac{1}{\mu B} \sum_{b=1}^{\mu B} \mathbb{1}(max(p_b') \geq \tau) \mathcal{H}(\hat{p}_b, p_b^s)$;
        {Contrastive Learning};
        $z_b^w = g \circ f(\mathcal{A}_w(u_b))$, $z_b^s = g \circ f(\mathcal{A}_s(u_b))$;
        $\mathbf{Q} = \{Q_i\}_{i=1}^C$, $Q_i \leftarrow (z_b, y_b)$;
        $\mathbf{C} = \{\hat{c}_i\}_{i=1}^N \leftarrow$ the average of the features $\mathbf{Q}$;
        $s_b^w = \mathrm{sort}(p_b^w, \text{descending}) = \{p_{\sigma(1)}^w, \dots, p_{\sigma(K)}^w, \dots, p_{\sigma(C)}^w\}$,
          $\{\sigma(1), \dots, \sigma(C)\}$ represents the descending order;
        $\hat{\mathbf{Q}} = \{\hat{Q}_i\}_{i=1}^N$, $\hat{\mathbf{C}} = \{\hat{c}_i\}_{i=1}^N$ $\leftarrow$the average of the features $\hat{\mathbf{Q}}$;
        $q_b^w = \frac{<z_b^w, \hat{\mathbf{C}}>}{\|z_b^w\| \|\hat{\mathbf{C}}\|} / T_{proto}$, $q_b^s = \frac{<z_b^s, \hat{\mathbf{C}}>}{\|z_b^s\| \|\hat{\mathbf{C}}\|} / T_{proto}$;
        $\hat{q}_b = argmax(q_b^w)$;
        $\mathcal{L}_{ctr} = \frac{1}{\mu B} \sum_{b=1}^{\mu B} \mathcal{H}(\hat{q}_b, q_b^s)$;
        $\mathcal{L}_{total} = \mathcal{L}_{sup} + \lambda_{un}\mathcal{L}_{un} + \lambda_{ctr}\mathcal{L}_{ctr}$;
        update $f(\cdot)$, $h(\cdot)$, $g(\cdot)$ with AdamW to minimize $\mathcal{L}_{total}$;
    end
end
```

Algorithm 1: CRMSP algorithm

4. Experiments

4.1. Datasets and compared methods

4.1.1. Datasets

FUNSD [33] is a comprehensive collection of real, fully annotated, scanned forms with 149 samples for training and 50 samples for testing. The documents are noisy and vary widely in appearance, making form understanding a challenging task. The proposed dataset can be used for various tasks, including text detection, optical character recognition, spatial layout analysis, and entity. CORD [34] is typically utilized for receipt KIE, which consists of 800/100/100 receipts for training/validation/testing. The dataset consists of thousands of Indonesian receipts, which contain images and box/text annotations for OCR, and multi-level semantic labels for parsing. The proposed dataset can be used to address various OCR and parsing tasks.

We construct long-tailed versions of CIFAR10 (CIFAR10-LT), CIFAR100 (CIFAR100-LT), and STL10 (STL10-LT) separately. N_c represents the number of examples in class c for labeled data and unlabeled data. γ_l and γ_u are the imbalance ratios for labeled data and unlabeled data. The number of examples for each class except the head class is based on the formula $N_c = N_1 \cdot \gamma_l^{-\frac{c-1}{C-1}}$. It is important to note that

Table 1

Evaluation results with 5% and 10% of labeled training samples on two KIE benchmarks based on LayoutLMv2 and LayoutLMv3. The best result is in **bold**, the second-best result is in underline.

Base model	Method	FUNSD						CORD					
		5%			10%			5%			10%		
		P	R	F1	P	R	F1	P	R	F1	P	R	F1
LayoutLMv2	Fully-supervised	77.91	81.38	79.61	77.91	81.38	79.61	95.81	95.02	95.41	95.81	95.02	95.41
	FixMatch [5]	61.42	70.55	65.67	65.27	73.36	69.08	83.96	85.36	84.66	87.01	87.47	87.24
	Dash [6]	62.71	74.51	**68.10**	64.77	70.65	67.58	77.99	82.64	80.25	87.47	88.53	88.00
	FlexMatch [7]	59.14	75.66	66.39	62.85	72.40	67.29	84.60	84.60	84.60	86.84	87.17	87.01
	SimMatch [8]	63.29	70.85	66.86	70.06	77.72	73.69	82.83	83.40	83.11	88.21	89.21	88.71
	FreeMatch [9]	64.06	70.65	67.19	68.32	74.46	71.26	82.79	84.60	83.69	88.81	89.28	89.05
	CRMSP(Ours)	63.39	71.85	67.36	71.49	78.63	**74.89**	84.47	84.98	**84.73**	90.46	90.19	**90.33**
LayoutLMv3	Fully-supervised	77.74	80.33	79.01	77.74	80.33	79.01	93.09	93.51	93.30	93.09	93.51	93.30
	FixMatch [5]	57.40	67.46	62.02	62.74	73.87	67.85	82.73	83.17	82.95	86.72	87.25	86.98
	Dash [6]	53.53	63.34	58.02	61.91	67.41	64.54	82.11	83.47	82.78	83.77	86.87	85.29
	FlexMatch [7]	60.04	70.54	64.87	70.85	73.77	72.28	80.06	82.42	81.22	86.10	86.49	86.30
	SimMatch [8]	64.78	66.68	65.72	69.40	77.40	**73.18**	82.07	82.57	82.32	87.72	87.85	87.78
	FreeMatch [9]	60.37	72.88	66.04	67.17	71.14	69.10	82.06	82.87	82.46	87.08	89.06	88.06
	CRMSP(Ours)	63.89	71.98	**67.69**	67.68	74.91	71.12	82.41	84.15	**83.27**	91.51	91.09	**91.30**

within each N_c, examples in class c are arranged in descending order (i.e., $N_1 \geq \cdots \geq N_C$).

4.1.2. Compared methods

Our approach is compared with both classical and imbalanced SSL methods. Classical SSL methods include FixMatch [5], Dash [6], FlexMatch [7], SimMatch [8] and FreeMatch [9]. Imbalanced SSL methods include DARP [30], CReST [28] and DASO [14].

4.2. Experiments settings

Our experiments are conducted on NVIDIA Tesla V100 GPU. For KIE datasets, the split ratios (i.e., the proportions of labeled data) are 5% and 10%, and the batch size of labeled data is 4, μ is set to 1.0. To validate the effectiveness of our proposed SSL approach, we use Transformer-based models such as LayoutLMv2 and LayoutLMv3 as base models. For all methods, we employ the Adam optimizer with a fixed learning rate of 1e–5. We take precision, recall and f1-score as our evaluation metrics. Referring to [31], we set weak augmentation to none and strong augmentation to random swap, which means randomly swapping two words in the sentence n times. In training, temperature hyperparameter T_{proto} is set to 1.0. The EMA model is used for testing with a momentum factor 0.999. For two unsupervised supervised losses, both λ_{un} and λ_{ctr} are set to 0.1, τ is set to 0.95. For both LayoutLMv2 and LayoutLMv3, K was fine-tuned with a setting of 5.

For CV datasets, we train the CIFAR10-LT/CIFAR100-LT and STL10-LT on the Wide ResNet-28-2 [30] with 1.5M parameters for 250k iterations. The optimizer is SGD with a learning rate of 0.03 and a weight decay of 5e–4. We set the training epoch to 256 and the batch size to 64. The imbalance ratio includes the following settings: $\gamma = \gamma_l = \gamma_u = 100$, $\gamma = \gamma_l = \gamma_u = 10$ and $\gamma_l = 10, \gamma_u : unknown$. The number of examples for the head class of labeled data N_1 is set to $\{500, 1500, 50, 150, 450\}$, and the number of examples for the head class of unlabeled data M_1 is set to $\{4000, 3000, 400, 300, 100k\}$.

4.3. Results

4.3.1. Results on FUNSD and CORD

To validate the effectiveness of CRMSP, we perform experiments on FUNSD and CORD for the token classification task. Table 1 shows comparative results with 5% and 10% labeled samples based on LayoutLMv2 and LayoutLMv3. For all methods, we observe that the f1-score increases as the ratio of labeled data increases. In contrast, CRMSP improves the f1-score in the vast majority of experimental settings. Based on LayoutLMv3, it works particularly well on the CORD with 10% labeled data and achieves 4.32% and 3.24% f1-score gain compared with FixMatch and suboptimal method FreeMatch, respectively. On the FUNSD, our proposed CRMSP achieved an improvement of f1-score that were 5.81% and 1.20% higher compared to FixMatch and the suboptimal model SimMatch based on LayoutLMv2, respectively.

This indicates that CRMSP can more effectively utilize labeled data to reduce model bias under long-tailed distribution. Furthermore, we observe that our method even achieves an f1-score of 91.30%, which is close to the f1-score of 93.30% achieved by the fully-supervised LayoutLMv3 on the KIE task, while using only 10% of the labeled data.

4.3.2. Results on CIFAR10/100-LT and STL10-LT

To illustrate the generalization of CRMSP, we also conducted experiments on the CIFAR10/100-LT and STL10-LT, as shown in Table 2. We consider rebalancing biased pseudo-labels by matching (e.g., $\gamma = \gamma_l = \gamma_u$) or mismatching (e.g., $\gamma_l = 10, \gamma_u : unknown$) distributions between imbalanced labeled and unlabeled data ($\mathcal{X}$ and $\mathcal{U}$) in Table 2. When $\gamma_l = \gamma_u$, we compare the proposed CRMSP with several classical (i.e., FixMatch [5]) and imbalanced (i.e., DARP [30], CReST [28] and DASO [14]) SSL baseline methods. In the supervised scenario, the performance is relatively constrained compared to other semi-supervised learning methods.

Notably, CRMSP demonstrates comparable or even superior performance across most settings, exhibiting substantial improvements compared to the baseline methods SimMatch and DARP. Compared to the baseline FixMatch and baseline DARP, the accuracy has increased by 0.04%–0.60% and 0.02%–17.63%, respectively.

Overall, Table 2 demonstrates that our proposed CRMSP is not only effective in mitigating imbalance issues in the SSL domain of KIE but also applicable in the CV domain.

4.3.3. Per-class performance

By comparing the top-1 accuracy of different methods on the STL10-LT across per class (see Table 3), we designate classes C0–C4 as the head classes and C5–C9 as the tail classes. In the head classes, it can be observed that Dash performs well on some head classes (C1, C2, C3), while our method achieves the second-best accuracy in C4 with 89.0%. In the tail classes, our method achieves an accuracy of 95.0% for C8, which is second-best compared to FreeMatch. Remarkably, our method CRMSP achieves the best accuracy of 73.8%, 90.5%, and 87.1% in C5, C6, and C9, respectively, representing improvements of 0.8%, 0.1%, and 2.8% over the second-best class SimMatch. This table highlights the enhancement provided by our method for tail classes in long-tailed distribution.

Table 2
Top-1 accuracy (%) on CIFAR10-LT, CIFAR100-LT and STL10-LT. The best result is in **bold**, the second-best result is in underline.

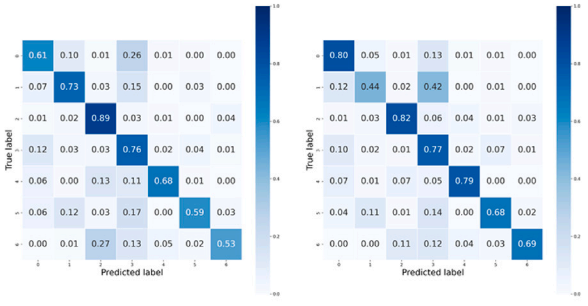
Method	Dataset	CIFAR10-LT		CIFAR100-LT		STL10-LT	
	imb_ratio	$\gamma = \gamma_l = \gamma_u = 100$		$\gamma = \gamma_l = \gamma_u = 10$		$\gamma_l = 10, \gamma_u$: *unknown*	
	#Label	$N_1 = 500$	$N_1 = 1500$	$N_1 = 50$	$N_1 = 150$	$N_1 = 150$	$N_1 = 450$
	#Unlabel	$M_1 = 4000$	$M_1 = 3000$	$M_1 = 400$	$M_1 = 300$	$M_1 = 100k$	$M_1 = 100k$
	Supervised	46.75	62.78	31.11	49.02	45.39	62.09
Classical SSL	FixMatch [5]	73.60	77.60	48.52	57.85	65.80	77.85
	Dash [6]	70.63	75.45	42.93	56.01	73.40	82.01
	FlexMatch [7]	62.18	73.69	38.99	53.88	82.75	85.55
	SimMatch [8]	**76.03**	78.68	47.30	57.86	82.85	85.75
	FreeMatch [9]	70.63	76.41	44.16	57.20	82.34	86.06
Imbalanced SSL	DARP [30]	75.92	78.62	**49.05**	57.88	65.38	75.70
	CReST [28]	71.21	75.74	44.37	55.77	63.58	71.70
	DASO [14]	71.12	76.79	48.35	57.71	68.37	79.03
	CRMSP(Ours)	71.38	**78.91**	44.53	**57.90**	**83.01**	**86.35**

Table 3
Per-class top-1 accuracy (%) on the balanced test dataset of STL10-LT ($\gamma_l = 10, \gamma_u$: *unknown*, $N_1 = 150, M_1 = 100k$). Our method shows a significant improvement in pseudo-labeling for tail classes. The best result is in **bold**, the second-best result is in underline.

Method	Class	0	1	2	3	4	5	6	7	8	9
	Supervised	86.1	61.0	81.5	49.4	38.8	23.8	42.9	18.5	29.0	23.0
Classical SSL	FixMatch [5]	73.8	4.7	0.2	48.2	80.4	53.9	0.2	1.1	42.1	38.1
	Dash [6]	94.1	**92.0**	**96.1**	**60.0**	86.3	70.5	83.8	50.0	69.0	32.3
	FlexMatch [7]	**95.4**	83.1	94.6	54.4	85.8	69.9	89.6	**80.3**	94.0	80.5
	SimMatch [8]	92.0	85.0	94.4	58.0	87.9	73.0	90.4	69.0	95.0	84.3
	FreeMatch [9]	94.5	84.3	94.1	46.3	**90.3**	67.8	88.9	77.6	**95.8**	84.0
Imbalanced SSL	DARP [30]	93.3	82.6	93.3	54.4	81.0	34.9	79.9	26.4	64.5	43.6
	CReST [28]	55.3	37.3	58.6	59.5	68.4	20.9	73.5	53.1	78.6	82.1
	DASO [14]	90.8	75.3	92.3	52.9	82.6	53.8	82.8	31.0	72.1	50.4
	CRMSP(Ours)	93.0	82.6	94.3	56.0	89.0	**73.8**	**90.5**	68.9	95.0	**87.1**

Table 4
Ablation study on FUNSD and CORD. "RP", "L_{ctr}" and "MP" mean Reweighting Pseudo-Labels, Contrastive Loss and Merged Prototypes, respectively. We use supervised loss as baseline.

#	RP	L_{ctr}	MP	FUNSD	CORD
0				66.56	84.05
1		✓		67.21	84.76
2		✓	✓	67.75	85.36
3	✓			68.96	87.56
4	✓	✓		69.91	88.08
5	✓	✓	✓	71.12	89.59



(a) w/o RP (b) w RP

Fig. 4. Confusion matrixes of the predictions on the test dataset of FUNSD.

4.4. Ablation study

To verify the effectiveness of each component of our proposed method, We conduct extensive ablation studies on the CORD. The results are shown in Table 4. And the evaluation metric for all experiments was the f1-score. We utilize LayoutLMv3 as the base model.

4.4.1. Effectiveness of reweighting pseudo-labels

To verify the effectiveness of RP, By comparing Experiment 2 and Experiment 5 in Table 4, we can observe that the RP improves the f1-score by 3.37% and 4.23% on the FUNSD and CORD. Table 5 We also compared the performance of tail classes without and with RP. It is found that by adding RP, there is a significant improvement in the results of the tail classes, especially for *num.sub_cnt* (0.42→0.78). Note that LayoutLMv2 is used as the base model.

To demonstrate the detailed effectiveness of our proposed RP, we present confusion matrixes of the predictions on the test dataset of FUNSD. As depicted in Fig. 4, the pseudo-labels for the tail classes without RP (e.g., C4, C5 and C6) are underestimated, while the accuracy between the pseudo-labels and the true labels for the head classes is higher. Our proposed RP improves the generation of more balanced pseudo-labels for tail classes, alleviating the issue of long-tailed distribution.

4.4.2. Effectiveness of contrastive loss

When incorporating contrastive loss, CRMSP can further boost the performance on all settings by another few points, resulting in 0.52% to 0.66% absolute accuracy improvement by comparing Experiment 3 and Experiment 4 in Table 4.

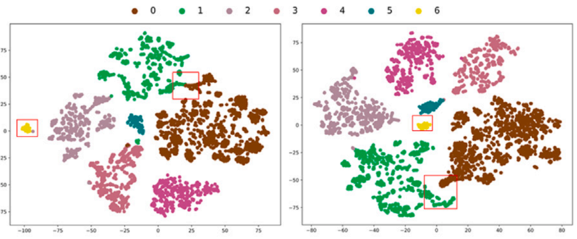
4.4.3. Effectiveness of merged prototypes

Comparing Experiment 4 and Experiment 5 in Table 4, we observed that f1-score improved by 1.21% and 1.51% on the FUNSD and CORD, respectively.

To demonstrate the effectiveness of our proposed MP, we present the comparison of t-SNE visualization of unlabeled data. As shown in Fig. 5, the MP helps the tail class (e.g., C6) to be separated from the confusion class and better clustering is achieved. Other confusing features (e.g., C0 and C1) are also clustered more compactly. Fig. 5 effectively promotes intra-class compactness and inter-class separability of unlabeled tail classes in feature space.

Table 5
The influence of the RP on the f1-score of tail classes on the CORD, “support” denotes the sample size.

Class	menu.sub_price	menu.sub_cnt	total.creditcardprice	sub_total.service_price	menu.num	sub_total.etc	sub_total.discount_price
Support	20	17	16	12	11	8	7
w/o RP	0.72	0.42	0.46	0.88	0.40	0.33	0.73
w RP	**0.88**	**0.78**	**0.73**	**0.96**	**0.74**	**0.62**	**0.88**



(a) w/o MP

(b) w MP

Fig. 5. Comparison of t-SNE visualization of unlabeled data on the FUNSD.

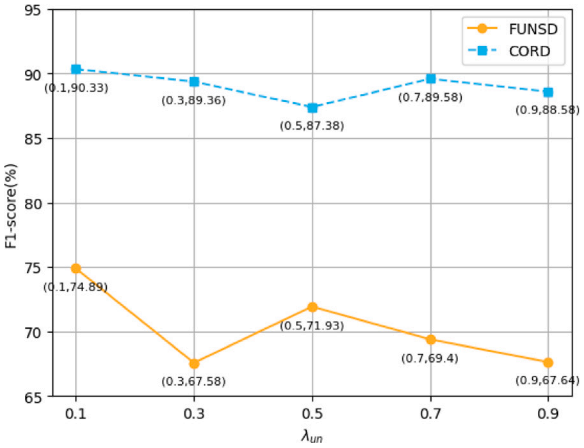


Fig. 6. The influence of different λ_{un} on the f1-score on the FUNSD and CORD.

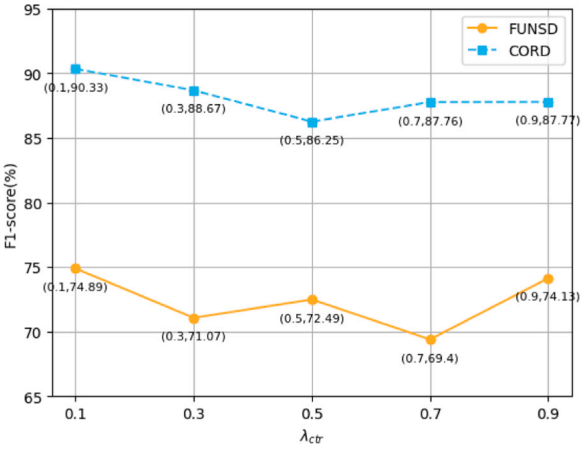


Fig. 7. The influence of different λ_{ctr} on the f1-score on the FUNSD and CORD.

4.4.4. Ablation study on λ_{un}

In Fig. 6, we study the effect of the temperature hyper-parameter λ_{un} to compute the weights for unsupervised loss described in Eq. (18). We empirically find that, for both FUNSD and CORD, $\lambda_{un} = 0.1$ shows the best performance.

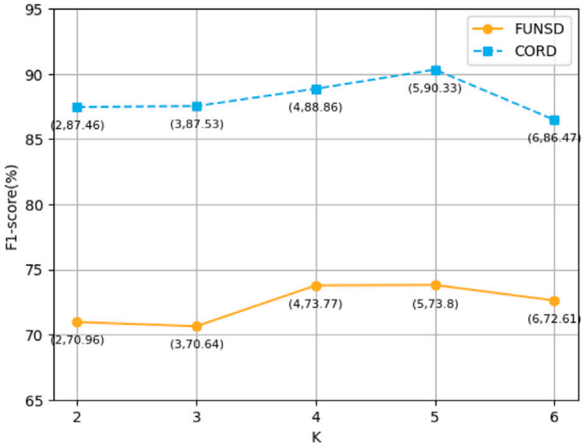


Fig. 8. The influence of different K on the f1-score on the FUNSD and CORD.

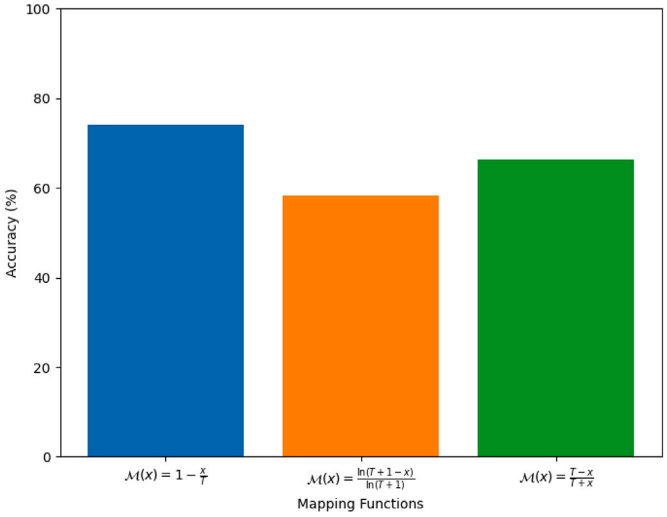


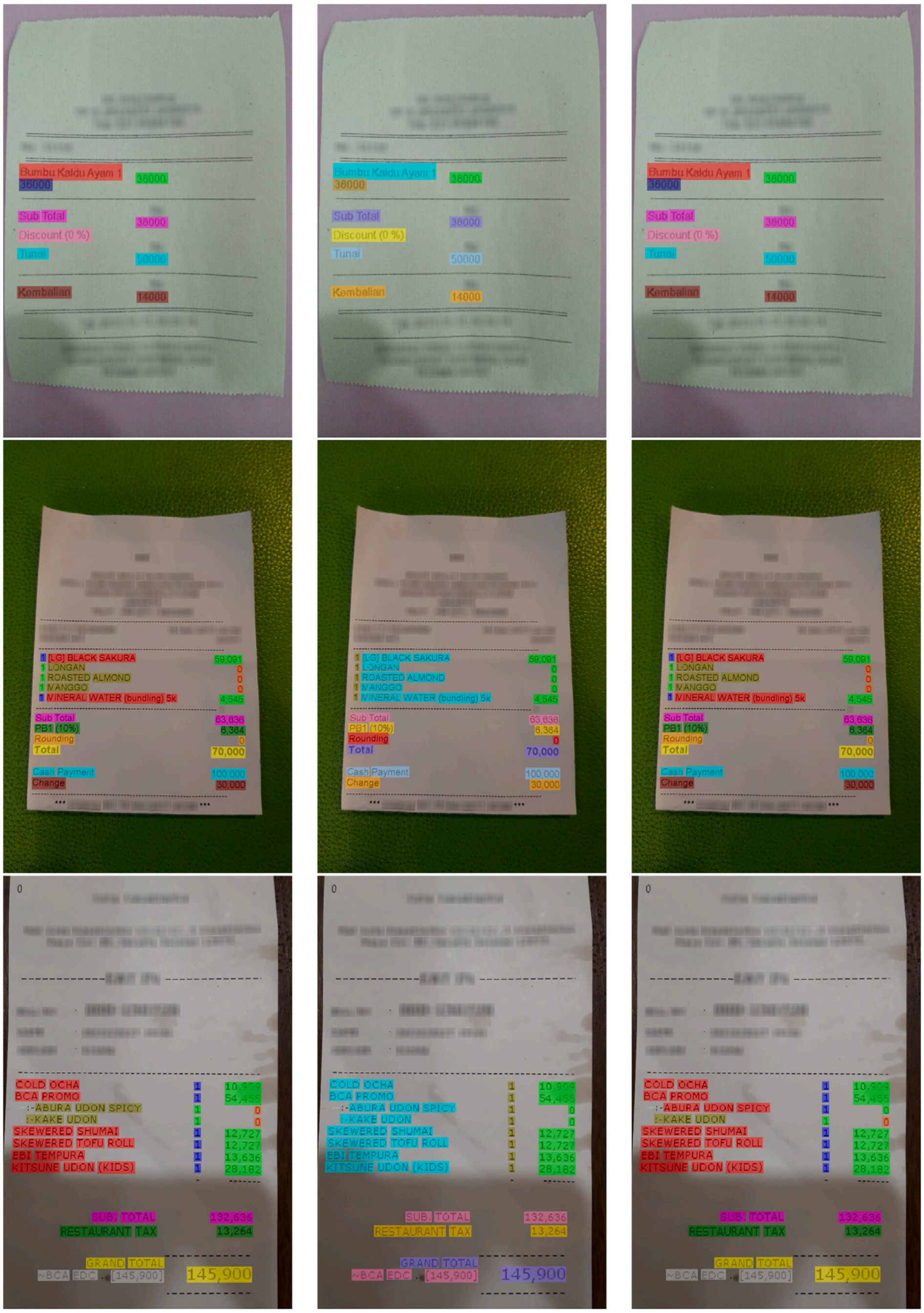
Fig. 9. Mapping function.

4.4.5. Ablation study on λ_{ctr}

In Fig. 7, we investigate the impact of the temperature hyper-parameter λ_{ctr} on computing the weights for the contrastive loss described in Eq. (18). $\lambda_{ctr} = 0.1$ yields the optimal performance on the FUNSD and CORD.

4.4.6. Ablation study on K

The influence of different K on the f1-score on the FUNSD and CORD is illustrated in Fig. 8. We notice that $K = 5$ provides the best f1-score among all tested values. When K is set to a small value, prototypes for some tail samples lack representation. Comparing these tail samples with non-representative prototypes results in semantic pseudo-labels that push the feature in the wrong direction in the feature space, leading to classification errors. On the other hand, if K is set too large, although the new sample features are effectively separated from classes not belonging to this super-class, the super feature range extends



(a) Ground-truth (b) FixMatch (c) CRMSP

Fig. 10. Example output of Ground-truth, FixMatch, and CRMSP for CORD.

far beyond the range of variations in tail features, causing internal confusion within the super-class.

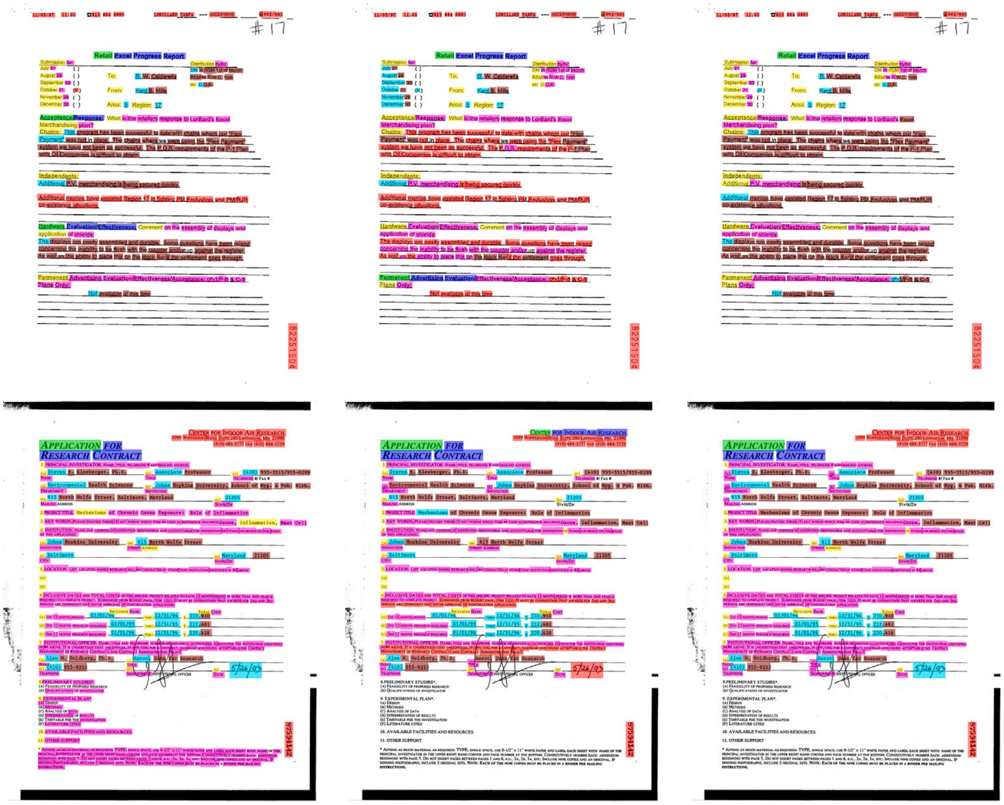
4.4.7. Ablation study on $\mathcal{M}(\cdot)$

To demonstrate the effectiveness of the mapping function $\mathcal{M}(\cdot)$, we designed three different mapping functions: (1) a linear function: $\mathcal{M}(x) = 1 - \frac{x}{T}$, (2) a concave function: $\mathcal{M}(x) = \frac{\ln(T+1-x)}{\ln(T+1)}$, and (3) a convex function: $\mathcal{M}(x) = \frac{T-x}{T+x}$. We conducted experiments under our settings, and the results are shown in Fig. 9. We can observe that when $T = 1$, the mapping function $\mathcal{M}(x) = 1 - \frac{x}{T}$ improves the accuracy in pseudo-labeling.

The function we designed is monotonically decreasing in the range of 0 to 1. In addition to the specific functions we selected for the experiment, other linear, convex, and concave decreasing functions can also be customized for different ranges. The parameter T adjusts the steepness of the function's decline.

4.4.8. Case study

We present the output of samples for Ground-truth, FixMatch, and CRMSP on both the CORD and FUNSD. On the CORD, as depicted in Fig. 10, the tail class *sub_total.discount_price* in the ground-truth is



(a) Ground-truth (b) FixMatch (c) CRMSP

Fig. 11. Example output of Ground-truth, FixMatch, and CRMSP for FUNSD.

incorrectly classified as *total.total_etc* by FixMatch. This misclassification is corrected by our proposed CRMSP approach. The tail classes *menu.sub_cnt* and *menu.sub_nm* are erroneously associated with their respective classes *menu.cnt* and *menu.nm*, but our proposed CRMSP method adeptly distinguishes between them.

On the FUNSD, as shown in Fig. 11, FixMatch misclassifies B-QUESTION and I-QUESTION as B-ANSWER and I-ANSWER, respectively. However, CRMSP correctly identifies these tail classes.

5. Conclusion

In this paper, we propose a novel semi-supervised approach for key information extraction with Class-Rebalancing and Merged Semantic Pseudo-Labeling (CRMSP). Firstly, the Class-Rebalancing Pseudo-Labeling (CRP) module is proposed to directly rebalance pseudo-labels with a reweighting factor, increasing attention to tail classes. Secondly, the Merged Semantic Pseudo-Labeling (MSP) module is proposed to achieve intra-class compactness and inter-class separability of unlabeled tail classes in feature space by assigning samples to Merged Prototypes (MP). We even achieved close to fully-supervised learning in the semi-supervised setting. Extensive experimental results have demonstrated the proposed CRMSP surpasses other state-of-the-art methods on five benchmarks. Our findings suggest that the proposed approach can obtain high-quality pseudo-labels from a larger amount of unlabeled data, which provides a good solution for semi-supervised learning.

CRediT authorship contribution statement

Qi Zhang: Writing – original draft, Validation, Software, Methodology, Conceptualization. **Yonghong Song:** Funding acquisition. **Pengcheng Guo:** Validation, Conceptualization. **Yangyang Hui:** Validation.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Yonghong Song reports financial support was provided by Xi'an Jiaotong University. Yonghong Song reports a relationship with Xi'an Jiaotong University that includes: employment, funding grants, non-financial support, speaking and lecture fees, and travel reimbursement. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work is supported by the National Natural Science Foundation of China, Grant Number: 62350083 and the Key R&D Plan of Shaanxi Province, Program No. 2023-YBGY-029.

Data availability

The data that has been used is confidential.

References

[1] Y. Xu, Y. Xu, T. Lv, L. Cui, F. Wei, G. Wang, Y. Lu, D. Florencio, C. Zhang, W. Che, et al., Layoutlmv2: Multi-modal pre-training for visually-rich document understanding, 2020, arXiv preprint arXiv:2012.14740.

[2] Y. Huang, T. Lv, L. Cui, Y. Lu, F. Wei, Layoutlmv3: Pre-training for document ai with unified text and image masking, in: ACM MM, 2022, pp. 4083–4091.

[3] P. Guo, Y. Song, Y. Deng, K. Xie, M. Xu, J. Liu, H. Ren, DCMAI: A dynamical cross-modal alignment interaction framework for document key information extraction, IEEE TCSVT (2023).

[4] X.J. Zhu, Semi-supervised learning literature survey, 2005.

[5] K. Sohn, D. Berthelot, N. Carlini, Z. Zhang, H. Zhang, C.A. Raffel, E.D. Cubuk, A. Kurakin, C.-L. Li, Fixmatch: Simplifying semi-supervised learning with consistency and confidence, NeurIPS 33 (2020) 596–608.

[6] Y. Xu, L. Shang, J. Ye, Q. Qian, Y.-F. Li, B. Sun, H. Li, R. Jin, Dash: Semi-supervised learning with dynamic thresholding, in: ICML, PMLR, 2021, pp. 11525–11536.

[7] B. Zhang, Y. Wang, W. Hou, H. Wu, J. Wang, M. Okumura, T. Shinozaki, Flexmatch: Boosting semi-supervised learning with curriculum pseudo labeling, NeurIPS 34 (2021) 18408–18419.

[8] M. Zheng, S. You, L. Huang, F. Wang, C. Qian, C. Xu, Simmatch: Semi-supervised learning with similarity matching, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 14471–14481.

[9] Y. Wang, H. Chen, Q. Heng, W. Hou, Y. Fan, Z. Wu, J. Wang, M. Savvides, T. Shinozaki, B. Raj, et al., Freematch: Self-adaptive thresholding for semi-supervised learning, 2022, arXiv preprint arXiv:2205.07246.

[10] Q. Xie, Z. Dai, E. Hovy, T. Luong, Q. Le, Unsupervised data augmentation for consistency training, NeurIPS 33 (2020) 6256–6268.

[11] Y. Zhang, B. Kang, B. Hooi, S. Yan, J. Feng, Deep long-tailed learning: A survey, IEEE TPAMI (2023).

[12] X. Huang, C. Zhu, W. Chen, Semi-supervised domain adaptation via prototype-based multi-level learning, 2023, arXiv preprint arXiv:2305.02693.

[13] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, A. Joulin, Unsupervised learning of visual features by contrasting cluster assignments, NeurIPS 33 (2020) 9912–9924.

[14] Y. Oh, D.-J. Kim, I.S. Kweon, Daso: Distribution-aware semantics-oriented pseudo-label for imbalanced semi-supervised learning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 9786–9796.

[15] Y. Xu, M. Li, L. Cui, S. Huang, F. Wei, M. Zhou, Layoutlm: Pre-training of text and layout for document image understanding, in: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2020, pp. 1192–1200.

[16] Q. Xie, M.-T. Luong, E. Hovy, Q.V. Le, Self-training with noisy student improves imagenet classification, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 10687–10698.

[17] A. Tarvainen, H. Valpola, Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results, Adv. Neural Inf. Process. Syst. 30 (2017).

[18] S. Laine, T. Aila, Temporal ensembling for semi-supervised learning, 2016, arXiv preprint arXiv:1610.02242.

[19] H. Fang, W. Deng, Y. Zhong, J. Hu, Triple-GAN: Progressive face aging with triple translation loss, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, 2020, pp. 804–805.

[20] Z. Liu, J. Wang, Z. Liang, Catgan: Category-aware generative adversarial networks with hierarchical evolutionary learning for category text generation, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 34, 2020, pp. 8425–8432.

[21] D. Berthelot, N. Carlini, E.D. Cubuk, A. Kurakin, K. Sohn, H. Zhang, C. Raffel, Remixmatch: Semi-supervised learning with distribution alignment and augmentation anchoring, 2019, arXiv preprint arXiv:1911.09785.

[22] C.-W. Kuo, C.-Y. Ma, J.-B. Huang, Z. Kira, Featmatch: Feature-based augmentation for semi-supervised learning, in: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVIII 16, Springer, 2020, pp. 479–495.

[23] N.V. Chawla, K.W. Bowyer, L.O. Hall, W.P. Kegelmeyer, SMOTE: synthetic minority over-sampling technique, J. Artif. Intell. Res. 16 (2002) 321–357.

[24] B. Dong, P. Zhou, S. Yan, W. Zuo, LPT: Long-tailed prompt tuning for image classification, 2023.

[25] J. Gao, H. Zhao, Z. Li, D. Guo, Enhancing minority classes by mixing: an adaptative optimal transport approach for long-tailed classification, Adv. Neural Inf. Process. Syst. 36 (2024).

[26] M. Li, H. Zhikai, Y. Lu, W. Lan, Y.-m. Cheung, H. Huang, Feature fusion from head to tail for long-tailed visual recognition, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 38, 2024, pp. 13581–13589.

[27] J. Chen, B. Su, Transfer knowledge from head to tail: Uncertainty calibration under long-tailed distribution, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 19978–19987.

[28] C. Wei, K. Sohn, C. Mellina, A. Yuille, F. Yang, Crest: A class-rebalancing self-training framework for imbalanced semi-supervised learning, in: CVPR, 2021, pp. 10857–10866.

[29] M. Hyun, J. Jeong, N. Kwak, Class-imbalanced semi-supervised learning, 2020, arXiv preprint arXiv:2002.06815.

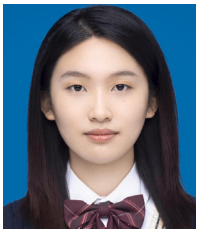
[30] J. Kim, Y. Hur, S. Park, E. Yang, S.J. Hwang, J. Shin, Distribution aligning refinery of pseudo-label for imbalanced semi-supervised learning, NeurIPS 33 (2020) 14567–14579.

[31] Y. Wang, H. Chen, Y. Fan, W. Sun, R. Tao, W. Hou, R. Wang, L. Yang, Z. Zhou, L.-Z. Guo, et al., Usb: A unified semi-supervised learning benchmark for classification, NeurIPS 35 (2022) 3938–3961.

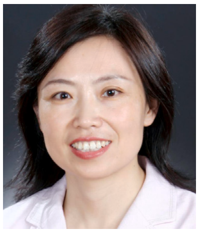
[32] O. Chapelle, B. Schlkopf, A. Zien, Semi-supervised learning (adaptive computation and machine learning), Mit Pr (2006) 2006.

[33] G. Jaume, H.K. Ekenel, J.-P. Thiran, Funsd: A dataset for form understanding in noisy scanned documents, in: ICDARW, Vol. 2, IEEE, 2019, pp. 1–6.

[34] S. Park, S. Shin, B. Lee, J. Lee, J. Surh, M. Seo, H. Lee, CORD: a consolidated receipt dataset for post-OCR parsing, in: Workshop on Document Intelligence At NeurIPS 2019, 2019.



Qi Zhang is pursuing her M.S. degree in Computer Science at Xi'an Jiaotong University, where she was to be advised by Yonghong Song. Prior to that, She did her B.S. in Automation at Nanjing Normal University. Her primary research focus is on key information extraction from complex documents based on pre-trained large language models.



YongHong Song (Member, IEEE) is currently a Professor with the School of Software Engineering, Xi'an Jiaotong University. Her current research interests include image and video content understanding, computer vision, pattern recognition, and intelligent systems. Many of her research findings have been applied in industry.

Pengcheng Guo received the M.S. degree in computer science and technology from Inner Mongolia University, China, in 2021. He is currently pursuing a Ph.D. degree in the School of Software Engineering, Xi'an Jiaotong University. His major research interests include document key information extraction, medical image segmentation, and multimedia analysis.

Yangyang Hui is currently pursuing the M.S. degree in-software engineering with Xi'an Jiaotong University. His research interests include natural language processing and document understanding.