IOCC: Aligning Semantic and Cluster Centers for Few-shot Short Text Clustering

Anonymous ACL submission

Abstract

In clustering tasks, it is essential to structure the feature space into clear, well-separated distributions. However, because short text representations have limited expressiveness, conventional methods struggle to identify cluster centers that truly capture each category's underlying semantics, causing the representations to be optimized in suboptimal directions. To address this issue, we propose **IOCC**, a novel few-shot contrastive learning method that achieves alignment between the cluster centers and the semantic centers. IOCC consists of two key modules: Interaction-enhanced Optimal Transport (**IEOT**) and Center-aware Contrastive Learning (**CACL**). Specifically, IEOT incorporates semantic interactions between individual samples into the conventional optimal transport problem, and generate pseudo-labels. Based on these pseudo-labels, we aggregate high-confidence samples to construct *pseudo-centers* that approximate the semantic centers. Next, CACL optimizes text representations toward their corresponding *pseudo-centers*. As training progresses, the collaboration between the two modules gradually reduces the gap between cluster centers and semantic centers. Therefore, the model will learn a high-quality distribution, improving clustering performance. Extensive experiments on eight benchmark datasets show that IOCC outperforms previous methods, achieving up to **7.34%** improvement on challenging Biomedical dataset and also excelling in clustering stability and efficiency. The code is available at: https://anonymous.4open.science/r/IOCC-C438.

1 Introduction

Short text clustering, which groups short texts into distinct clusters based on their semantic similarity, has broad applications in real-world domains such as chatbots (Kuhail et al., 2023), topic discovery (Murshed et al., 2023), and spam detection (Liu et al., 2024; Abkenar et al., 2023). A key factor in achieving high-quality clustering is determining the appropriate cluster center for each category, as this critically influences whether samples can be grouped according to their intrinsic semantic similarities (Bai et al., 2012). The ideal scenario is that the cluster center for each category precisely corresponds to the semantic center (i.e., the core or central concept that embodies the main meaning of the category) in the feature space. However, due to the lack of labeled samples and limitations in text representation quality, extracting the semantic center of each category remains a challenge (Fini et al., 2023). As illustrated in Figure 1(a), cluster centers often fail to align with the semantic centers, leading to suboptimal category aggregation.

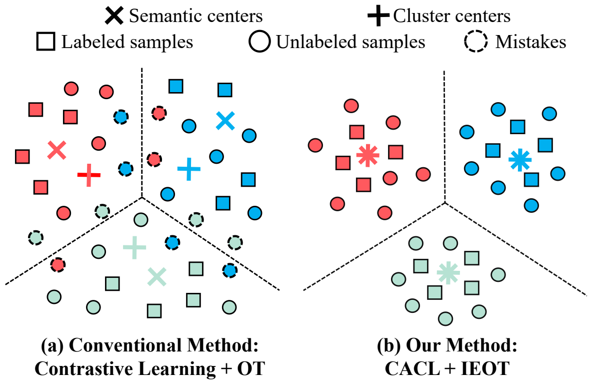

Figure 1: **Schematic Illustration of the Motivation.** (**a**) Previous works generate cluster centers that are misaligned with the underlying semantic centers. (**b**) In contrast, our method effectively aligns cluster centers with the semantic centers by constructing *pseudo-centers*, thereby facilitating a finer distribution.

Previously, Zheng et al. (2023); Li et al. (2024) proposed constructing pseudo-labels to assign preliminary category information to certain samples, allowing similar samples to gradually converge during the iterative process. However, the pseudo-labels generated using traditional optimal transport are limited to the global structure and ignore individual information, which reduces the accuracy of the pseudo-labels. On the other hand, to learn more discriminative and robust text representations, Zhang et al. (2021); Chen et al. (2020) introduced contrastive learning, which optimizes text representations by pulling positive pairs together and pushing negative pairs apart in the feature space. However, these method only consider instance-wise relationships, neglecting category-wise optimization, which causes samples that should belong to the same category to be pushed apart, affecting cluster quality (Wang and Isola, 2020).

In this work, we propose **IOCC**, a novel few-shot contrastive learning framework for short text clustering. The primary objective of this model is to pull the text representations toward the correct corresponding centers in the feature space. IOCC combines two key components: Interaction-enhanced Optimal Transport (**IEOT**) and Center-aware Contrastive Learning (**CACL**).

Specifically, (1) we incorporate similarity interactions between samples into the optimal transport (OT) framework, enabling IEOT to generate more reliable pseudo-labels. (2) We then combine minimal true labels with pseudo-labels to effectively design a *pseudo-center* to approximate the semantic center for each category. Next, CACL leverage these *pseudo-centers* as targets, pulling samples toward their corresponding *pseudo-center* while pushing them away from the others. As training progresses, the collaboration between the above two modules drives the *pseudo-centers* to gradually approach the true semantic centers, which in turn guides the text representations to move closer to them. Eventually, IOCC aligns the cluster centers with the semantic centers, yielding a more optimal distribution, as shown in Figure 1(b).

We demonstrate that IOCC achieves state-of-the-art performance on eight benchmark datasets. Notably, IOCC achieved the highest accuracy in all datasets, with improvements exceeding **7.34%** and **4.18%** on Biomedical and GoogleNews-T, respectively. Additionally, we show that our method exhibits faster convergence and more robust training compared to current methods. In summary, our main contributions are as follows:

(1) We propose a few-shot framework, IOCC, which integrates the following two key components, bridging the gap between the semantic and cluster centers. (2) We propose a novel optimal transport strategy, IEOT, which integrates semantic interactions between individual samples. It generates reliable pseudo-labels to help the few-shot labels uncover the true semantic centers of each category. (3) We propose a novel contrastive learning method, CACL, which aligns cluster centers with semantic centers by constructing *pseudo-centers* to guide the representation optimization. (4) IOCC shows state-of-the-art results on eight benchmark datasets. it also achieves faster convergence and better stability compared to previous methods.

2 Related Works

Short Text Clustering. Short text clustering is challenging due to the limited number of words in short texts. In recent years, deep joint clustering methods have become mainstream by integrating representation learning and clustering into a unified framework. Notable examples include SCCL (Zhang et al., 2021), which uses DEC (Xu et al., 2017) as the clustering objective and contrastive learning to guide representation learning. RSTC (Zheng et al., 2023) proposes the use of pseudo-labels to assist the model in learning sample representations and clustering. STSPL-SSC (Nie et al., 2024) is built on the RSTC method, using fewer labeled data to assist the pseudo-labeling process. COTC (Li et al., 2024) combines sentence-level and token-level information to achieve more efficient clustering.

Few-shot learning. Few-shot methods leverage a small amount of labeled data and a large collection of unlabeled data to train models. The most intuitive approach is Pseudo-labels (Lee et al., 2013), where a model trained on labeled data generates pseudo-labels for unlabeled examples, which are then added to the labeled set for the next iteration. However, hard labels easily exacerbate the classification bias of the training model (confirmation bias) (Arazo et al., 2020). To counteract this issue, researchers have shown benefits from soft labels and confidence thresholding (Arazo et al., 2020) as well as from different training strategies like co- and tri-training (Dong-DongChen and WeiGao, 2018; Nassar et al., 2021). In our research, we integrate optimal transport and pseudo-labeling methods to explore textual features and similarities, maximizing the guiding role of labeled information.

Contrastive Learning. As a promising paradigm of unsupervised learning, contrastive learning has lately achieved state-of-the-art performance in many fields (Grill et al., 2020). Contrastive learning aims to map data to a feature space where positive pairs are similar and negative pairs are dissimilar (Hadsell et al., 2006). Recently, Zhang et al. (2021) applies contrastive learning to short text clustering, upon which methods like Zheng et al. (2023); Nie et al. (2024); Li et al. (2024) and many others have introduced further improvements. The previous methods typically distribute the samples uniformly in feature space (Wang and Isola, 2020), whereas our approach further optimizes them by incorporating semantics, thereby achieving consistency and accuracy.

3 Method

IOCC is primarily attributed to two key factors: Interaction-enhanced Optimal Transport (**IEOT**) and Center-aware Contrastive Learning (**CACL**), as illustrated in Figure 2. Specifically, after samples pass through the Encoder and Classifier, IEOT processes their probability distributions to generate pseudo-labels. Subsequently, *pseudo-centers* are updated by aggregating high-confidence samples which can better represent the semantics of categories. CACL then enforces that each text representation in the feature space is contracted toward its corresponding *pseudo-center*. Eventually, *pseudo-centers* gradually converge toward the semantic centers, thereby achieving alignment between cluster centers and semantic centers.

3.1 Preliminaries

In our method, we train the model using M labeled samples and N unlabeled samples, where $N \gg M$. Following (Zhang et al., 2021), we apply the *contextual augmenter* (Shorten et al., 2021) to generate augmented data by inserting or substituting top-n suitable words of the input text. Given an unlabeled sample $\boldsymbol{x}_i^{u(0)}$ and a labeled sample $\boldsymbol{x}_i^{l(0)}$, their augmented versions are defined as $\{\boldsymbol{x}_i^{u(1)}, \boldsymbol{x}_i^{u(2)}\}$ and $\{\boldsymbol{x}_i^{l(1)}, \boldsymbol{x}_i^{l(2)}\}$, respectively. During training, mini-batches are constructed from labeled instances $\mathcal{X} = \{(\boldsymbol{x}_j^{l(0)}, y_j^l)\}_{j=1}^{B}$, and unlabeled instances $\mathcal{U} = \{(\boldsymbol{x}_i^{u(0)}\}_{i=1}^{\mu \cdot B}$. Here, B is the batch size of labeled data, μ is the ratio of unlabeled to labeled examples in each mini-batch, and y_j^l is the true label corresponding to the cluster $k \in \{1, \dots, K\}$. We denote the Encoder as $f(\cdot)$, followed by a Classifier network $g(\cdot)$ and a Projector network $h(\cdot)$. For each sample, the probability output of the Classifier is defined as $\boldsymbol{p}_i \in \mathbb{R}^K = g \circ f(\boldsymbol{x}_i)$. The projected representations from the Projector are defined as $\boldsymbol{z}_i \in \mathbb{R}^D = h \circ f(\boldsymbol{x}_i)$.

3.2 Interaction-enhanced Optimal Transport

Based on previous optimal transport (OT) methods (Zheng et al., 2023), IEOT incorporates a novel regularization term constructed using the semantic similarity between individual samples. By solving this novel OT problem, we can derive pseudo-labels that seamlessly combine the semantic interactions imposed by our regularization with the global structure captured by the standard OT formulation.

Given a batch of original unlabeled samples $\boldsymbol{X}^{u^{(0)}}$, we define the probability assignments as $\boldsymbol{P}^{u^{(0)}} \in \mathbb{R}^{\mu B \times K} = g \circ f(\boldsymbol{X}^{u^{(0)}})$. Then, pseudo-labels can be generated by solving the IEOT problem as follows:

$$\begin{aligned} &\min_{\boldsymbol{Q},\boldsymbol{b}} \langle \boldsymbol{Q},\boldsymbol{M}\rangle - \varepsilon_1 H(\boldsymbol{Q}) + \varepsilon_2 \Theta(\boldsymbol{b}) - \varepsilon_3 \langle \boldsymbol{S},\boldsymbol{Q}\boldsymbol{Q}^T\rangle \\ &\textit{s.t. } \boldsymbol{Q}\boldsymbol{1}_K = \boldsymbol{a},\ \boldsymbol{Q}^T\boldsymbol{1}_{\mu B} = \boldsymbol{b},\ \boldsymbol{Q} \geq 0,\ \boldsymbol{b}^T\boldsymbol{1}_K = 1, \end{aligned} \tag{1}$$

where $\boldsymbol{M} = -\log(\boldsymbol{P}^{u^{(0)}})$, $\boldsymbol{Q}$ is the transport matrix, $\langle \cdot, \cdot \rangle$ is the Frobenius inner product, ε_1, ε_2 and ε_3 are hyperparameters, $\boldsymbol{a} = \frac{1}{N}\boldsymbol{1}_{\mu B}$ is the sample distribution, and $\boldsymbol{b}$ is an unknown cluster distribution. $\boldsymbol{S}$ is the cosine similarity matrix of the probability assignment $\boldsymbol{P}^{u^{(0)}}$ defined as follows:

$$S_{ij} = \frac{\langle \boldsymbol{P}_{i:}^{u^{(0)}}, \boldsymbol{P}_{j:}^{u^{(0)}} \rangle}{\|\boldsymbol{P}_{i:}^{u^{(0)}}\|_2 \, \|\boldsymbol{P}_{j:}^{u^{(0)}}\|_2}, \tag{2}$$

where $\boldsymbol{P}_{i:}^{u^{(0)}}$ denote the i-th row vector of $\boldsymbol{P}^{u^{(0)}}$. Details of each term in Eq.(1) are as follows:

- $H(\boldsymbol{Q}) = -\langle \boldsymbol{Q}, \log\boldsymbol{Q} - 1\rangle$ is the entropy of the transport matrix $\boldsymbol{Q}$, which prevents $\boldsymbol{Q}$ from being sparse.
- $\Theta(\boldsymbol{b}) = \sum_{j=1}^{K} -b_j \log(b_j)$ is the entropy of the cluster probability $\boldsymbol{b}$, which encourages $\boldsymbol{b}$ to approach a uniform distribution. By adjusting the strength of this term, IEOT is suitable for various imbalanced datasets.
- $\langle \boldsymbol{S}, \boldsymbol{Q}\boldsymbol{Q}^T\rangle$ is the semantic regularization, which promotes the transport matrix $\boldsymbol{Q}$ to capture semantic similarity between samples. Specifically, this term encourages the transport vector $\boldsymbol{Q}_{i:}$ to be similar to $\boldsymbol{Q}_{j:}$ when the similarity S_{ij} is large. In other words, it ensures semantically similar samples produce similar transport vectors.

IEOT is a non-convex optimization problem. We propose to solve this problem by using the

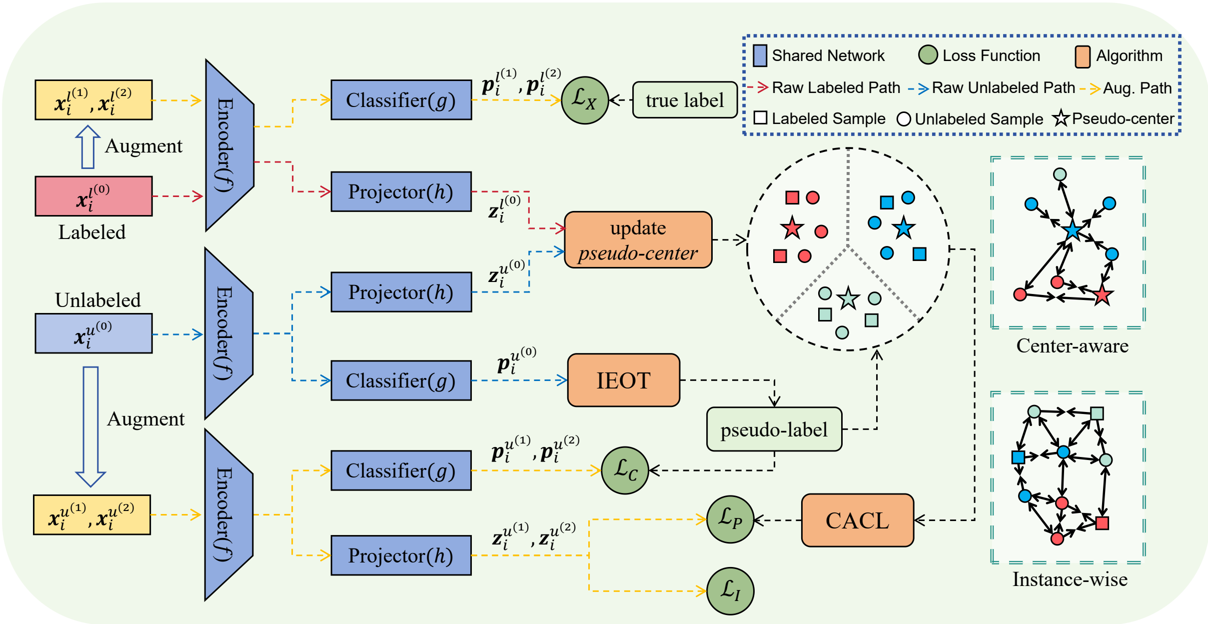

Figure 2: **Method Overview. IOCC** is mainly composed of two core components: Interaction-enhanced Optimal Transport (**IEOT**) and Center-aware Contrastive Learning (**CACL**).

Majorization-Minimization method which minimizes the objective function by iteratively minimizing its surrogate function (Hunter and Lange, 2004). Details of the solution are provided in Appendix A.1.

By solving the proposed IEOT problem, we obtain the transport matrix $\boldsymbol{Q}$, which not only serves as a probability assignment matrix reflecting the traditional OT's global sample-to-cluster structure but also encodes semantic interactions between individual samples. Finally, pseudo-label for the i-th sample $\hat{y}_i^u$ can be generated as follows:

$$\hat{y}_i^u = \arg\max_j Q_{ij}. \tag{3}$$

In other words, the pseudo-label for a given unlabeled sample corresponds to the cluster with the highest corresponding assignment probability.

3.3 Center-aware Contrastive Learning

After obtaining the pseudo-labels, we aim to promote well-clustered short text projections by attracting samples to their respective semantic centers while distancing them from the others. Therefore, we adopt a contrastive objective that utilizes *pseudo-centers* to approximate the semantic centers. *Pseudo-centers* are computed at the end of each iteration, based on the labeled and high-confidence pseudo-labeled samples identified from the previous iteration.

Specifically, we define a reliability indicator for each sample $\eta_i = \mathbb{1}(\max(\boldsymbol{p}_i^{u(0)}) \geq \tau)$ denoting if its max prediction exceeds the confidence threshold τ. Formally, let $\mathcal{I}_k^l = \{i|\forall \boldsymbol{x}_i^{l(0)} \in \mathcal{X}, y_i^l = k\}$ be the indices of labeled instances with true cluster k, and $\mathcal{I}_k^u = \{i|\forall \boldsymbol{x}_i^{u(0)} \in \mathcal{U}, \eta_i = 1, \hat{y}_i^u = k\}$ be the indices of the reliable unlabeled samples with hard pseudo-label k. The normalized *pseudo-center* $\boldsymbol{c}_k$ for cluster k can then be obtained as per:

$$\bar{\boldsymbol{c}}_k = \frac{\sum_{i \in \mathcal{I}_k^u \cup \mathcal{I}_k^l} \boldsymbol{z}_i}{|\mathcal{I}_k^u| + |\mathcal{I}_k^l|}, \quad \boldsymbol{c}_k = \frac{\bar{\boldsymbol{c}}_k}{||\bar{\boldsymbol{c}}_k||_2}. \tag{4}$$

In the following iteration, we minimize the following Center-aware Contrastive Learning (CACL) loss on unlabeled augmented samples:

$$\begin{aligned}\mathcal{L}_P = & -\frac{1}{\mu B}\sum_{i=1}^{\mu B} \log \frac{\exp(\cos(\boldsymbol{z}_i^{u(1)}, \boldsymbol{c}_{\hat{y}_i^u})/T_P)}{\sum_{k=1}^{K} \exp(\cos(\boldsymbol{z}_i^{u(1)}, \boldsymbol{c}_k)/T_P)} \\ & -\frac{1}{\mu B}\sum_{i=1}^{\mu B} \log \frac{\exp(\cos(\boldsymbol{z}_i^{u(2)}, \boldsymbol{c}_{\hat{y}_i^u})/T_P)}{\sum_{k=1}^{K} \exp(\cos(\boldsymbol{z}_i^{u(2)}, \boldsymbol{c}_k)/T_P)},\end{aligned} \tag{5}$$

where $\cos(\boldsymbol{z}_i^{u(1)}, \boldsymbol{c}_{\hat{y}_i^u})$ denotes the cosine similarity between $\boldsymbol{z}_i^{u(1)}$ and the *pseudo-center* $\boldsymbol{c}_{\hat{y}_i^u}$ corresponding to $\hat{y}_i^u$, with T_P meaning the temperature parameter. Consequently, *pseudo-centers* will gradually converge to the semantic centers, and samples from the same category will be more tightly distributed around the semantic center in the feature space, thereby enhancing clustering performance.

3.4 Instance-wise Contrastive Learning

To help the model capture finer details from the augmented samples, we also employ Instance-wise Contrastive Learning. For the i-th unlabeled sample in a batch, its augmented samples are regarded as a positive pair, while the other $2\mu B - 2$ pairs are considered negative. The loss function for the i-th sample is defined as follows:

$$l_i = -\log \frac{\delta(\boldsymbol{z}_i^{u(1)}, \boldsymbol{z}_i^{u(2)})}{\sum_{\substack{k=1\\k\neq i}}^{\mu B} (\delta(\boldsymbol{z}_i^{u(1)}, \boldsymbol{z}_k^{u(1)}) + \delta(\boldsymbol{z}_i^{u(1)}, \boldsymbol{z}_k^{u(2)}))} - \log \frac{\delta(\boldsymbol{z}_i^{u(2)}, \boldsymbol{z}_i^{u(1)})}{\sum_{\substack{k=1\\k\neq i}}^{\mu B} (\delta(\boldsymbol{z}_i^{u(2)}, \boldsymbol{z}_k^{u(1)}) + \delta(\boldsymbol{z}_i^{u(2)}, \boldsymbol{z}_k^{u(2)}))}. \tag{6}$$

Here $\delta(\boldsymbol{z}_i^{u(1)}, \boldsymbol{z}_i^{u(2)}) = \exp(\cos(\boldsymbol{z}_i^{u(1)}, \boldsymbol{z}_i^{u(2)})/T_I)$, T_I is a temperature parameter. The total loss is computed as follows:

$$\mathcal{L}_I = \frac{1}{\mu B} \sum_{i=1}^{\mu B} l_i. \tag{7}$$

3.5 Pseudo-label & Supervised Learning

Using the generated pseudo-labels, we compute the loss for unlabeled samples based on the model's prediction under augmentations, as follows:

$$\mathcal{L}_C = \frac{1}{\mu B} \sum_{i=1}^{\mu B} (\mathrm{CE}(\hat{y}_i^u, \boldsymbol{p}_i^{u(1)}) + \mathrm{CE}(\hat{y}_i^u, \boldsymbol{p}_i^{u(2)})), \tag{8}$$

where CE denotes the cross-entropy. Also, we apply a supervised classification loss over the labeled data:

$$\mathcal{L}_X = \frac{1}{B} \sum_{i=1}^{B} (\mathrm{CE}(y_i^l, \boldsymbol{p}_i^{l(1)}) + \mathrm{CE}(y_i^l, \boldsymbol{p}_i^{l(2)})). \tag{9}$$

Notably, Eq.(8) and Eq.(9) are acted on both two augmented versions.

3.6 Final Objective

We design a two-stage training procedure for IOCC. The first stage aims to obtain a good initial feature space, while the second stage focuses on optimizing the distribution using all the algorithms mentioned above. The overall loss function is:

$$\mathcal{L} = \begin{cases} \mathcal{L}_X + \mathcal{L}_C + \mathcal{L}_I & \text{if } iter < E_{first} \\ \mathcal{L}_X + \mathcal{L}_C + \mathcal{L}_I + \lambda \mathcal{L}_P & \text{if } iter \geq E_{first} \end{cases} \tag{10}$$

where $iter$ is the number of training iterations, λ is a balancing hyperparameter, and E_{first} is the first stage iterations. By integrating the above components, the model learns a high-quality feature space distribution, leading to more accurate and stable clustering results. Algorithm 2 in Appendix E.1 describes the training process of IOCC.

4 Experiments

4.1 Datasets

We conducted experiments using eight benchmark datasets: **AgNews**, **StackOverflow**, **Biomedical**, **SearchSnippets**, **GoogleNews-TS**, **GoogleNews-T**, **GoogleNews-S**, and **Tweet**. A summary of the key characteristics and detailed information of these datasets are provided in Table 1 and Appendix E.2, respectively.

Datasets	S	N	L	R
AgNews	8000	4	23	1
SearchSnippets	12340	8	18	7
StackOverflow	20000	20	8	1
Biomedical	20000	20	13	1
GoogleNews-TS	11109	152	8	143
GoogleNews-T	11109	152	6	143
GoogleNews-S	11109	152	22	143
Tweet	2472	89	22	249

Table 1: **Key Information of Datasets.** "S" represents the dataset size; "N" is the number of categories; "L" is the average sentence length; "R" is the size ratio of the largest to the smallest category.

4.2 Experiment Settings

We implement our model using PyTorch (Paszke et al., 2019) and employ *bge-base-en-v1.5* in the Sentence Transformers library as the Encoder (Chen et al., 2024). Under our few-shot definition, we use 1% of the samples as labeled samples if **S/N** $> 1\%$ according to Table 1, otherwise we use only 1 sample per dataset as labeled samples. All parameters of our model are optimized using the Adam optimizer (Kingma, 2014). The learning rate of the Encoder is 5×10^{-6}, while the other networks is 5×10^{-4}. We use Accuracy (ACC) and Normalized Mutual Information (NMI) to evaluate the model. Definitions of the metrics and detailed settings are in Appendix E.3 and Appendix E.4.

4.3 Baselines

We compare **IOCC** with several latest short text clustering approaches. **SCCL** (Zhang et al., 2021)

	AgNews		SearchSnippets		StackOverflow		Biomedical	
	ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI
SCCL	83.10	61.96	79.90	63.78	70.83	69.21	42.49	39.16
RSTC	84.24	62.45	80.10	69.74	83.30	74.11	48.40	40.12
BGE-M3	87.89	66.67	75.59	60.7	84.66	82.21	51.25	46.05
MIST	89.47	70.25	76.72	67.69	79.65	78.59	39.15	34.66
STSPL-SSC	<u>89.92</u>	<u>71.66</u>	81.04	65.46	86.74	<u>82.54</u>	47.43	42.49
COTC	87.56	67.09	<u>90.32</u>	<u>77.09</u>	<u>87.78</u>	79.19	<u>53.20</u>	<u>46.09</u>
IOCC	**90.28**	**72.22**	**90.44**	**77.15**	**90.38**	**82.74**	**60.54**	**48.81**
Improvement	**+0.36**	**+0.56**	**+0.12**	**+0.06**	**+2.6**	**+0.20**	**+7.34**	**+2.72**
	GoogleNews-TS		**GoogleNews-T**		**GoogleNews-S**		**Tweet**	
	ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI
SCCL	82.51	93.01	69.01	85.10	73.44	87.98	73.10	86.66
RSTC	83.27	93.15	72.27	87.39	79.32	89.40	75.20	87.35
BGE-M3	72.97	91.81	68.28	87.52	69.89	89.01	64.64	87.42
MIST	<u>90.63</u>	**96.42**	78.80	89.31	82.14	90.86	<u>91.75</u>	**95.12**
STSPL-SSC	84.41	94.32	81.01	91.11	82.30	91.18	79.59	88.02
COTC	90.50	<u>96.33</u>	<u>83.53</u>	<u>92.07</u>	<u>86.10</u>	**93.49**	91.33	<u>95.09</u>
IOCC	**92.92**	95.90	**87.71**	**92.39**	**87.64**	<u>92.79</u>	**92.11**	94.63
Improvement	**+2.29**	**-0.52**	**+4.18**	**+0.32**	**+1.54**	**-0.7**	**+0.36**	**-0.49**

Table 2: **Experimental Results.** Clustering performance of IOCC and baselines are presented on eight benchmarks. The results of baselines are quoted from (Zheng et al., 2023; Li et al., 2024; Kamthawee et al., 2024; Nie et al., 2024). We bold the **best result**, underline the <u>runner-up</u>.

employs contrastive learning to refine representations and obtains the clustering results using the DEC algorithm (Xie et al., 2016). **RSTC** (Zheng et al., 2023) constructs pseudo-labels using adaptive optimal transport to assist the model in training neural networks for clustering. **MIST** (Kamthawee et al., 2024) enhances clustering by maximizing the mutual information between representations at both the sequence and token levels. **STSPL-SSC** (Nie et al., 2024) extends RSTC by incorporating additional labeled data and leveraging the information from these labels to guide the effectiveness of pseudo-labels. **COTC** (Li et al., 2024) introduces a Co-Training Clustering framework that effectively combines BERT and TFIDF features to generate a high-quality feature space for clustering.

Additionally, to measure the performance of the Encoder, we include **BGE-M3** experiments, which apply k-means directly to the output of the BGE-M3 model. Further analysis of the same Encoder on other baselines are conducted in Appendix B.3.

4.4 Main Results

The clustering results for both baseline models and IOCC are summarized in Table 2. From the results, we can find that: (1) The traditional contrastive learning method **SCCL** and the **RSTC** method with the introduction of OT, due to the complexity of the datasets, did not yield good results. (2) Directly incorporating k-means in **BGE-M3** cannot achieve good clustering results. (3) **MIST** and **COTC** allow the model to learn more features, and thus performed second only to IOCC on some datasets. However, they still struggled to address the challenges posed by complex semantics. (4) **STSPL-SSC**, by introducing semi-supervised learning, demonstrated good performance; nevertheless, the information it could learn still fell short of our method, so did its performance. (5) Obviously, **IOCC** consistently outperforms previous methods across all datasets. Notably, IOCC achieves superior clustering accuracy, particularly on more challenging datasets such as Biomedical, GoogleNews-T, and StackOverflow. The two components in IOCC cooperate with each other to extract scarce information, achieving a more clear and well-separated distribution in the feature space, which is essential for achieving such outstanding results. In the following sections, numerous ex-

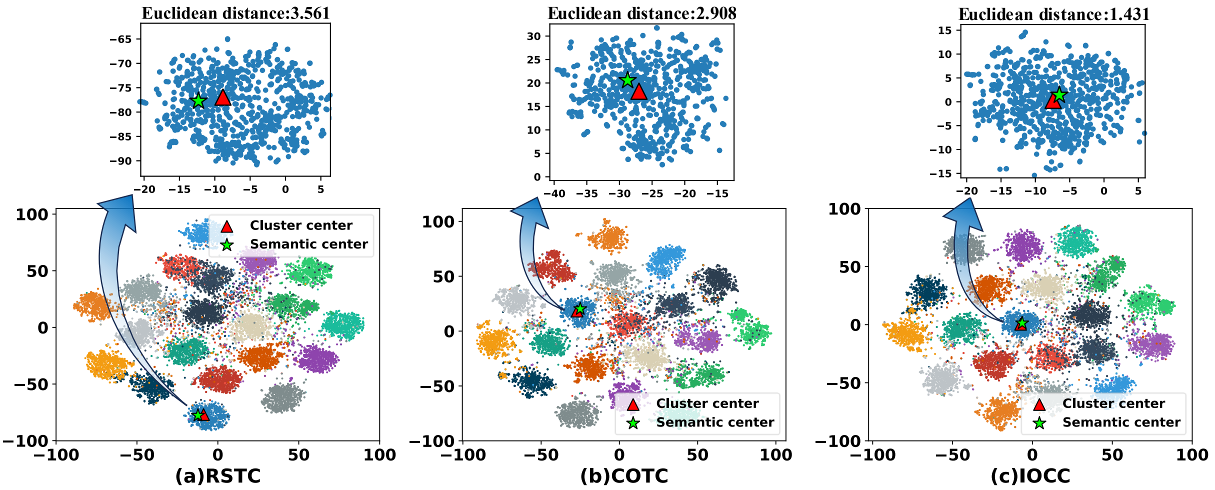

Figure 3: **Comparison of the Alignment Between Semantic Center and Cluster Center.** The semantic center is calculated as the mean embedding of the keywords that describe the category, whereas the cluster center is the average embedding of all samples within the category. Each color indicates a truth category.

periments will be presented to further validate the accuracy and stability of our model.

4.5 Semantic Alignment Visualization

We use t-SNE visualization and Euclidean distances to verify whether IOCC achieves semantic alignment. Specifically, we chose a representative category from the StackOverflow dataset — the category named "Matlab", where all samples consist of sentences describing "matlab". We generated a *Word Cloud* to identify the keywords in this category, and used the average embedding of these keywords to represent the semantic center of the category (the list of keywords includes: "matlab", "functions", "matrices", "visualization", "programming", "scripts", and "optimization."). The visualization of the cluster center and the semantic center is shown in Figure 3, compared to other models, IOCC achieves the best alignment between the cluster centers and the semantic centers. It reveals that our method accurately determine the cluster centers in the feature space.

Furthermore, we can observe that the feature space distribution obtained by IOCC is more consistent and compact. A more detailed comparison of the representation visualizations is provided in Appendix B.2.

4.6 The Comparison of Model Stability

To validate the stability of our model, we used multiple different random seeds to observe variations in model performance. Specifically, we conducted experiments on the AgNews and SearchSnippets datasets, with random seeds ranging from 0 to 10. To ensure a fair comparison, all experiments uniformly use BGE-M3 as Encoder. The results are shown in Figure 4. From it, we can find that: (1) RSTC demonstrates high stability but performs poorly on the imbalanced SearchSnippets dataset. (2) COTC exhibits lower stability. (3) IOCC achieves the highest performance while maintaining strong stability, demonstrating the robustness and generalizability of our model.

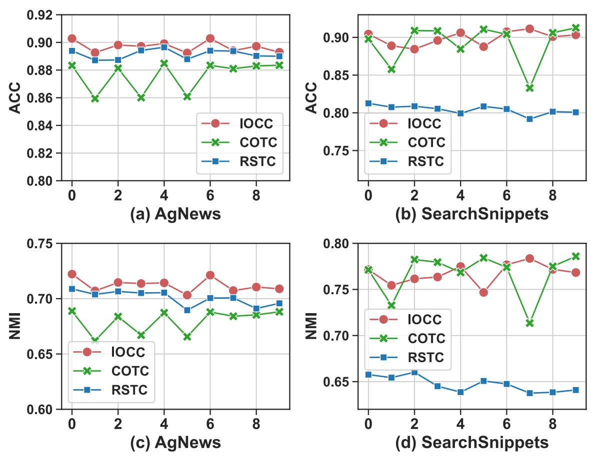

Figure 4: **Comparison of Stability**. The x-axis representing the random seeds we used.

4.7 Ablation Study

To demonstrate that each proposed module in IOCC contributes to the outstanding performance, we conducted ablation experiments on eight datasets, as shown in Table 3. The experimental results demonstrate that the model performance significantly decreases regardless of which module we remove from IOCC. When CACL is removed, rely-

Modules	Agn	Sea	Sta	Bio	GN-TS	GN-T	GN-S	Twe	δ
$-$(IEOT&CACL)	86.50	81.70	86.74	49.40	81.13	64.79	73.04	73.21	-11.94
$-$(IEOT)	87.41	84.24	88.10	53.51	82.77	67.23	74.86	75.32	-9.82
$-$(CACL)	88.79	87.51	89.33	58.17	91.47	86.42	86.48	90.41	-1.68
IOCC	90.28	90.44	90.38	60.54	92.92	87.71	87.64	92.11	0

Table 3: **Ablation Results**. $-(*)$ denotes the respective module is removed. δ is the average improvement over IOCC.

Labeled count	Agn	Sea	Sta	Bio	GN-TS	GN-T	GN-S	Twe
1 or 1%	90.28	90.44	90.38	60.54	92.92	87.71	87.64	92.11
2 or 2%	90.41	91.13	90.83	63.51	94.21	89.1	90.86	94.7
5 or 5%	91.13	92.35	91.22	69.43	95.02	90.35	91.17	95.23
10 or 10%	91.65	93.25	91.96	73.41	96.25	93.09	92.84	98.46

Table 4: **The Impact of Varying the Number of Labeled Samples.** Note that, when (**S/N**) $\leq 1\%$, if the required labeled samples for a class exceed its available samples, the available number of samples in that class is used instead.

ing solely on IEOT to generate pseudo-labels fails to optimize the distribution in the feature space. On the other hand, when IEOT is removed, CACL cannot utilize reliable pseudo-labels, causing the failure in learning the correct information. Only when each part of the model collaborates with the others can the best performance be achieved.

4.8 The Impact of Labeled Data Quantity

Furthermore, we conduct experiments by varying the number of labeled samples to 1 or 1%, 2 or 2%, 5 or 5%, 10 or 10%, where "1 or 1%" means that: we use 1% of the samples as labeled if (**S/N**) > 1% according to Table 1, and we use only 1 sample per category as labeled if (**S/N**) $\leq 1\%$. The results are presented in Table 4. We can observe that the performance increases with the number of labeled samples. In few-shot settings, IOCC already achieves state-of-the-art results, and as more labeled data is collected, the model's performance continues to improve. This demonstrates that IOCC can effectively be applied in real-world scenarios. Finally, we construct the labeled data using the "1 or 1%" setting, which offers the highest cost-effectiveness.

4.9 In-depth Analysis

In addition to the experiments mentioned above, we conducted more supplementary experiments to further verify the capabilities of IOCC:

(1) We recorded how the number of predicted clusters are changing over iterations in Appendix B.1, showing that our model can effectively combat clustering degeneracy. (2) Since each baseline model uses a different Encoder, we converted baseline models to the same Encoder (BGE-M3 and SBERT) for comparison. The results provided in the Appendix B.3, it can be observed that, regardless of whether the Encoder is the same or not, our model outperforms all other models. (3) Due to the current scarcity of semi-supervised methods in the field of short text clustering, we incorporated labeled data into recent high-performance models in the training process. As can be seen from the Appendix B.4, few-shot scenario will not directly enhance the performance of the baselines, and IOCC still outperforms these models comprehensively. (4) We conducted hyperparameter analysis experiments including $\varepsilon_1, \varepsilon_2, \varepsilon_3$ and λ, and analyzed the impact of these hyperparameters in Appendix D. (5) We recorded the computation budget with previous models, as shown in Appendix C. Our model strikes a balance between performance and efficiency, making it the most cost-effective solution.

5 Conclusion

This paper presents a novel approach, **IOCC**, for few-shot short text clustering, which combines Interaction-enhanced Optimal Transport (**IEOT**) and Center-aware Contrastive Learning (**CACL**). The former significantly improved the accuracy of pseudo-labels by exploiting the interaction between samples, while the latter aligning the cluster centers with the semantic centers by constructing *pseudo-centers* and pulling samples towards them. Extensive experiments demonstrate that IOCC consistently outperforms existing state-of-the-art techniques, showing significant improvements in clustering accuracy and stability.

6 Limitations

Despite the promising results, there are some limitations to our method. (1) The performance slightly depends on the quality and representativeness of the labeled data. So the future work will focus on how to derive labeled data in a cost-effective way like using LLMs. (2) The pseudo-labeling process, while effective, can still introduce errors, particularly in noisy or ambiguous data. Therefore, exploring a method for generating more accurate pseudo-labels is also a key focus in the future.

References

Sepideh Bazzaz Abkenar, Mostafa Haghi Kashani, Mohammad Akbari, and Ebrahim Mahdipour. 2023. Learning textual features for twitter spam detection: A systematic literature review. volume 228, page 120366. Elsevier.

Eric Arazo, Diego Ortego, Paul Albert, Noel E O'Connor, and Kevin McGuinness. 2020. Pseudo-labeling and confirmation bias in deep semi-supervised learning. In *2020 International joint conference on neural networks (IJCNN)*, pages 1–8. IEEE.

Liang Bai, Jiye Liang, Chuangyin Dang, and Fuyuan Cao. 2012. A cluster centers initialization method for clustering categorical data. *Expert Systems with Applications*, 39(9):8022–8029.

Stephen Boyd and Lieven Vandenberghe. 2004. *Convex optimization*. Cambridge university press.

Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024. Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. In *Annual Meeting of the Association for Computational Linguistics*.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR.

W Dong-DongChen and ZH WeiGao. 2018. Tri-net for semi-supervised deep learning. In *Proceedings of twenty-seventh international joint conference on artificial intelligence*, pages 2014–2020.

Enrico Fini, Pietro Astolfi, Karteek Alahari, Xavier Alameda-Pineda, Julien Mairal, Moin Nabi, and Elisa Ricci. 2023. Semi-supervised learning made simple with self-supervised clustering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3187–3197.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. 2020. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284.

Raia Hadsell, Sumit Chopra, and Yann LeCun. 2006. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE computer society conference on computer vision and pattern recognition (CVPR'06)*, volume 2, pages 1735–1742. IEEE.

David R. Hunter and Kenneth Lange. 2004. A tutorial on mm algorithms. *The American Statistician*, 58(1):30–37.

Krissanee Kamthawee, Can Udomcharoenchaikit, and Sarana Nutanong. 2024. Mist: mutual information maximization for short text clustering. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11309–11324.

Diederik P Kingma. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Mohammad Amin Kuhail, Nazik Alturki, Salwa Alramlawi, and Kholood Alhejori. 2023. Interacting with educational chatbots: A systematic review. *Education and Information Technologies*, 28(1):973–1018.

Dong-Hyun Lee et al. 2013. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, volume 3, page 896. Atlanta.

Zetong Li, Qinliang Su, Shijing Si, and Jianxing Yu. 2024. Leveraging BERT and TFIDF Features for Short Text Clustering via Alignment-Promoting Co-Training. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 14897–14913.

Tianrui Liu, Shaojie Li, Yushan Dong, Yuhong Mo, and Shuyao He. 2024. Spam detection and classification based on distilbert deep learning algorithm. *Applied Science and Engineering Journal for Advanced Research*, 3(3):6–10.

Belal Abdullah Hezam Murshed, Suresha Mallappa, Jemal Abawajy, Mufeed Ahmed Naji Saif, Hasib Daowd Esmail Al-Ariki, and Hudhaifa Mohammed Abdulwahab. 2023. Short text topic modelling approaches in the context of big data: taxonomy, survey, and analysis. volume 56, pages 5133–5260. Springer.

Islam Nassar, Samitha Herath, Ehsan Abbasnejad, Wray Buntine, and Gholamreza Haffari. 2021. All labels are not created equal: Enhancing semi-supervision via label grouping and co-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7241–7250.

Wenhua Nie, Lin Deng, Chang-Bo Liu, JialingWei JialingWei, Ruitong Han, and Haoran Zheng. 2024. STSPL-SSC: Semi-Supervised Few-Shot Short Text Clustering with Semantic text similarity Optimized Pseudo-Labels. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 12174–12185, Bangkok, Thailand. Association for Computational Linguistics.

Christos H Papadimitriou and Kenneth Steiglitz. 1998. *Combinatorial optimization: algorithms and complexity*. Courier Corporation.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32.

Xuan-Hieu Phan, Le-Minh Nguyen, and Susumu Horiguchi. 2008. Learning to classify short and sparse text & web with hidden topics from large-scale data collections. In *Proceedings of the 17th international conference on World Wide Web*, pages 91–100.

Md Rashadul Hasan Rakib, Norbert Zeh, Magdalena Jankowska, and Evangelos Milios. 2020. Enhancement of short text clustering by iterative classification. In *Natural Language Processing and Information Systems: 25th International Conference on Applications of Natural Language to Information Systems, NLDB 2020, Saarbrücken, Germany, June 24–26, 2020, Proceedings 25*, pages 105–117. Springer.

Connor Shorten, Taghi M Khoshgoftaar, and Borko Furht. 2021. Text data augmentation for deep learning. *Journal of big Data*, 8(1):101.

Tongzhou Wang and Phillip Isola. 2020. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *International conference on machine learning*, pages 9929–9939. PMLR.

Junyuan Xie, Ross Girshick, and Ali Farhadi. 2016. Unsupervised deep embedding for clustering analysis. In *International conference on machine learning*, pages 478–487. PMLR.

Jiaming Xu, Bo Xu, Peng Wang, Suncong Zheng, Guanhua Tian, and Jun Zhao. 2017. Self-taught convolutional neural networks for short text clustering. *Neural Networks*, 88:22–31.

Jianhua Yin and Jianyong Wang. 2016. A model-based approach for text clustering with outlier detection. In *2016 IEEE 32nd International Conference on Data Engineering (ICDE)*, pages 625–636. IEEE.

Dejiao Zhang, Feng Nan, Xiaokai Wei, Shangwen Li, Henghui Zhu, Kathleen McKeown, Ramesh Nallapati, Andrew Arnold, and Bing Xiang. 2021. Supporting clustering with contrastive learning. *arXiv preprint arXiv:2103.12953*.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28.

Xiaolin Zheng, Mengling Hu, Weiming Liu, Chaochao Chen, and Xinting Liao. 2023. Robust Representation Learning with Reliable Pseudo-labels Generation via Self-Adaptive Optimal Transport for Short Text Clustering. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10493–10507.

A Hyper-efficient Solution for IEOT

A.1 Formulation of the Solution

As mentioned in Section 3.2, the IEOT problem is formulated as:

$$\min_{\boldsymbol{Q},\boldsymbol{b}} \langle \boldsymbol{Q}, \boldsymbol{M} \rangle - \varepsilon_1 H(\boldsymbol{Q}) + \varepsilon_2 \Theta(\boldsymbol{b}) - \varepsilon_3 \langle \boldsymbol{S}, \boldsymbol{Q}\boldsymbol{Q}^T \rangle$$
$$\textit{s.t.}\ \ \boldsymbol{Q}\mathbf{1}_K = \boldsymbol{a},\ \boldsymbol{Q}^T\mathbf{1}_{\mu B} = \boldsymbol{b},\ \boldsymbol{Q} \geq 0,\ \boldsymbol{b}^T\mathbf{1}_K = 1, \tag{11}$$

where $\boldsymbol{M} = -\log(\boldsymbol{P}^{u^{(0)}})$, $\langle \cdot, \cdot \rangle$ represents the Frobenius inner product, ε_1 and ε_2 are balancing hyperparameters, $\boldsymbol{a} = \frac{1}{\mu B}\mathbf{1}_{\mu B}$, $H(\boldsymbol{Q}) = -\langle \boldsymbol{Q}, \log(\boldsymbol{Q}) - 1 \rangle$, and $\Theta(\boldsymbol{b}) = \sum_{j=1}^{K} -b_j \log(b_j)$ is the entropy of the cluster probability assignments $\boldsymbol{b}$.

The IEOT incorporates a complex quadratic semantic regularization term, which cannot be solved directly using traditional OT methods. To address IEOT, we propose integrating the Lagrange multiplier algorithm (Zheng et al., 2023) into the Majorization-Minimization method to solve IEOT. The proposed Majorization-Minimization method is iteratively minimizes the objective function in Eq.(11). In the i-th ($i \geq 1$) iteration, the Taylor expansion with the constant term and the linear term to approximate $\langle \boldsymbol{S}, \boldsymbol{Q}\boldsymbol{Q}^T \rangle$ are as follows:

$$T(\boldsymbol{S}, \boldsymbol{Q}) = \langle (\boldsymbol{S} + \boldsymbol{S}^T)\boldsymbol{Q}_{i-1}, \boldsymbol{Q} - \boldsymbol{Q}_{i-1} \rangle + \langle \boldsymbol{S}, \boldsymbol{Q}_{i-1}\boldsymbol{Q}_{i-1}^T \rangle \tag{12}$$

, in which $\frac{\partial \langle \boldsymbol{S}, \boldsymbol{Q}\boldsymbol{Q}^T \rangle}{\partial \boldsymbol{Q}} = (\boldsymbol{S} + \boldsymbol{S}^T)\boldsymbol{Q}$ is used.

When replacing the $\langle \boldsymbol{S}, \boldsymbol{Q}\boldsymbol{Q}^T \rangle$ in the objective function with its Taylor approximation in Eq.(12), one can get the following optimization problem:

$$\min_{\boldsymbol{Q},\boldsymbol{b}} \langle \boldsymbol{Q}, \boldsymbol{M} \rangle - \varepsilon_1 H(\boldsymbol{Q}) + \varepsilon_2 \Theta(\boldsymbol{b}) - T(\boldsymbol{S}, \boldsymbol{Q})$$
$$\textit{s.t.}\ \ \boldsymbol{Q}\mathbf{1}_K = \boldsymbol{a},\ \boldsymbol{Q}^T\mathbf{1}_{\mu B} = \boldsymbol{b},\ \boldsymbol{Q} \geq 0,\ \boldsymbol{b}^T\mathbf{1}_K = 1, \tag{13}$$

The objective function in Eq.(13) is a surrogate function for the objective function in Eq.(11). To prove this claim, define

$$g(\boldsymbol{Q}, \boldsymbol{b}) = \langle \boldsymbol{Q}, \boldsymbol{M} \rangle - \varepsilon_1 H(\boldsymbol{Q}) + \varepsilon_2 \Theta(\boldsymbol{b}), \quad (14)$$

the objective function in Eq.(11) is

$$f(\boldsymbol{Q}, \boldsymbol{b}) = g(\boldsymbol{Q}, \boldsymbol{b}) - \varepsilon_3 \langle \boldsymbol{S}, \boldsymbol{Q}\boldsymbol{Q}^T \rangle, \quad (15)$$

and the objective function in Eq.(13) is

$$s(\boldsymbol{Q}, \boldsymbol{b}) = g(\boldsymbol{Q}, \boldsymbol{b}) - T(\boldsymbol{S}, \boldsymbol{Q}). \quad (16)$$

$f(\boldsymbol{Q}, \boldsymbol{b})$ and $s(\boldsymbol{Q}, \boldsymbol{b})$ satisfy the following two conditions:

$$\text{Condition 1: } f(\boldsymbol{Q}_{i-1}, \boldsymbol{b}_{i-1}) = s(\boldsymbol{Q}_{i-1}, \boldsymbol{b}_{i-1}) \quad (17)$$

$$\text{Condition 2: } f(\boldsymbol{Q}, \boldsymbol{b}) \leq s(\boldsymbol{Q}, \boldsymbol{b}), \quad (18)$$

Condition 1 is straightforward, while Condition 2 is based on the concavity of $\langle \boldsymbol{S}, \boldsymbol{Q}\boldsymbol{Q}^T \rangle$ w.r.t. $\boldsymbol{Q}$, such that the following inequality holds (Boyd and Vandenberghe, 2004):

$$\begin{aligned} -\langle \boldsymbol{S}, \boldsymbol{Q}\boldsymbol{Q}^T \rangle \leq & - \langle (\boldsymbol{S} + \boldsymbol{S}^T)\boldsymbol{Q}_{i-1}, \boldsymbol{Q} - \boldsymbol{Q}_{i-1} \rangle \\ & - \langle \boldsymbol{S}, \boldsymbol{Q}_{i-1}\boldsymbol{Q}_{i-1}^T \rangle. \end{aligned} \quad (19)$$

Based on these two conditions, $f(\boldsymbol{Q}, \boldsymbol{b})$ is a surrogate function for $s(\boldsymbol{Q}, \boldsymbol{b})$ (Hunter and Lange, 2004). One can solve the problem in Eq.(11) iteratively, and in each iteration the problem in Eq.(13) is solved. In the i-th iteration, with $\boldsymbol{Q}_{i-1}$ available, the objective function in Eq.(13) can be rewritten as follows:

$$\begin{aligned} & \langle \boldsymbol{Q}, \boldsymbol{M} \rangle - \varepsilon_1 H(\boldsymbol{Q}) + \varepsilon_2 \Theta(\boldsymbol{b}) - \varepsilon_3 T(\boldsymbol{S}, \boldsymbol{Q}) \\ & = \langle \boldsymbol{Q}, \boldsymbol{M} - \varepsilon_3 (\boldsymbol{S} + \boldsymbol{S}^T)\boldsymbol{Q}_{i-1} \rangle - \varepsilon_1 H(\boldsymbol{Q}) \\ & \quad + \Theta(\boldsymbol{b}) + D, \end{aligned} \quad (20)$$

in which $D = \varepsilon_3 \langle (\boldsymbol{S} + \boldsymbol{S}^T)\boldsymbol{Q}_{i-1}, \boldsymbol{Q}_{i-1} \rangle - \varepsilon_3 \langle \boldsymbol{S}, \boldsymbol{Q}_{i-1}\boldsymbol{Q}_{i-1}^T \rangle$ is a constant.

Therefore, the optimization problem in Eq.(13) can be rewritten as follows:

$$\begin{aligned} \min_{\boldsymbol{Q}, \boldsymbol{b}} \quad & \langle \boldsymbol{Q}, \widetilde{\boldsymbol{M}} \rangle - \varepsilon_1 H(\boldsymbol{Q}) + \varepsilon_2 \Theta(\boldsymbol{b}) \\ \text{s.t. } \quad & \boldsymbol{Q}\mathbf{1}_K = \boldsymbol{a}, \ \boldsymbol{Q}^T\mathbf{1}_{\mu B} = \boldsymbol{b}, \ \boldsymbol{Q} \geq 0, \ \boldsymbol{b}^T\mathbf{1}_K = 1, \end{aligned} \quad (21)$$

with $\widetilde{\boldsymbol{M}} = -\log(\boldsymbol{P}^{(0)}) - \varepsilon_3 (\boldsymbol{S} + \boldsymbol{S}^T)\boldsymbol{Q}_{i-1}$.

Then, we adopt the Lagrangian multiplier algorithm to solve Eq.(21):

$$\begin{aligned} \min_{\boldsymbol{Q}, \boldsymbol{b}} & \langle \boldsymbol{Q}, \widetilde{\boldsymbol{M}} \rangle - \varepsilon_1 H(\boldsymbol{Q}) + \varepsilon_2 \Theta(\boldsymbol{b}) - \boldsymbol{f}^T(\boldsymbol{Q}\mathbf{1}_K - \boldsymbol{a}) \\ & - \boldsymbol{g}^T(\boldsymbol{Q}^T\mathbf{1}_{\mu B} - \boldsymbol{b}) - h(\boldsymbol{b}^T\mathbf{1}_K - 1), \end{aligned} \quad (22)$$

where $\boldsymbol{f}$, $\boldsymbol{g}$ and h are all Lagrangian multipliers. Taking the partial derivative of Eq.(22) with respect to $\boldsymbol{Q}$, one can obtain:

$$Q_{ij} = \exp\left(\frac{f_i + g_j - \widetilde{M}_{ij}}{\varepsilon_1}\right) > 0. \quad (23)$$

Eq.(23) is a function of each element in $\boldsymbol{f}$ and $\boldsymbol{g}$. Next, we first fix $\boldsymbol{b}$, and update f_i and g_j. Due to the fact that $\boldsymbol{Q}\mathbf{1}_K = \boldsymbol{a}$, one can get:

$$\begin{aligned} \sum_{j=1}^{K} Q_{ij} &= \sum_{j=1}^{K} \exp\left(\frac{f_i + g_j - \widetilde{M}_{ij}}{\varepsilon_1}\right) \\ &= \exp\left(\frac{f_i}{\varepsilon_1}\right) \sum_{j=1}^{K} \exp\left(\frac{g_j - \widetilde{M}_{ij}}{\varepsilon_1}\right) \\ &= a_i, \end{aligned} \quad (24)$$

where K represents the number of clusters in the dataset. Further, one can obtain:

$$\exp\left(\frac{f_i}{\varepsilon_1}\right) = \frac{a_i}{\sum_{j=1}^{K} \exp\left(\frac{g_j - \widetilde{M}_{ij}}{\varepsilon_1}\right)}. \quad (25)$$

Taking the logarithm of both sides and multiplying by ε_1, one can obtain:

$$f_i = \varepsilon_1 \ln a_i - \varepsilon_1 \ln \sum_{j=1}^{K} \exp\left(\frac{g_j - \widetilde{M}_{ij}}{\varepsilon_1}\right). \quad (26)$$

Similar to the above derivation, from $\boldsymbol{Q}^T\mathbf{1}_{\mu B} = \boldsymbol{b}$, one can obtain:

$$g_j = \varepsilon_1 \ln b_j - \varepsilon_1 \ln \sum_{i=1}^{\mu B} \exp\left(\frac{f_i - \widetilde{M}_{ij}}{\varepsilon_1}\right). \quad (27)$$

We can observe that g_j is an unknown variable in Eq.(26), while f_i is an unknown variable in Eq.(27). Since f_i and g_j are functions of each other, making it infeasible to directly solve for their exact values. Thus, we employ an iterative approach to update and work out it.

Then, we fix $\boldsymbol{f}$ and $\boldsymbol{g}$, and update $\boldsymbol{b}$. Specifically, take the partial derivative of the optimization problem Eq.(22) on the variable $\boldsymbol{b}$, one can obtain:

$$\varepsilon_2(\log(b_j) + 1) + g_j - h = 0, \quad (28)$$

by solving formula Eq.(28), one can get:

$$b_j(h) = \exp\left(\frac{h - g_j - \varepsilon_2}{\varepsilon_2}\right). \quad (29)$$

Taking Eq.(29) back to the original constraint $\boldsymbol{b}^T\mathbf{1}_K = 1$, the formula is defined as below:

$$(\boldsymbol{b}(h))^T\mathbf{1}_K = \sum_{j=1}^{K}\exp(\frac{h-g_j-\varepsilon_2}{\varepsilon_2}) = 1, \quad (30)$$

by extracting the scalar part, one can obtain:

$$\exp(\frac{h}{\varepsilon_2})\sum_{j=1}^{K}\exp(\frac{-g_j-\varepsilon_2}{\varepsilon_2}) = 1, \quad (31)$$

by solving Eq.(31), one can get:

$$h = -\varepsilon_2\log\left(\sum_{j=1}^{K}\exp(\frac{-g_j-\varepsilon_2}{\varepsilon_2})\right), \quad (32)$$

where h is the root of Eq.(30), Then, we can obtain $\boldsymbol{b}$ by Eq.(29).

Overall, through iteratively updating the Eq.(26), (27) and (29), we can get the transport matrix $\boldsymbol{Q}$ on Eq.(23). We show the iterative optimization process for solving Eq.(21) using the Lagrange multiplier algorithm in Algorithm 1.

Algorithm 1 The pseudo-code for solving IEOT

Input: Probability matrix $\boldsymbol{P}^{(0)}$; marginal constraints $\boldsymbol{a}$; semantic similarity matrix $\boldsymbol{S}$; constraints weights ε_1, ε_2 and ε_3.
Output: Transport matrix $\boldsymbol{Q}$.
Procedure:

```
Initialize b_0 randomly and perform normalization so that b_0^T 1 = 1
Initialize Q_0 = a b_0^T.
for i = 1 to T_1 do
    M = −log(P^(0)) − ε_3(S + S^T) Q_{i−1}.
    Initialize f and g randomly.
    Initialize h = 1.
    for i=1 to T_2 do
        Fix b, update f and g by Eq.(26) and (27), respectively.
        Fix f and g, update b by Eq.(29) and (32).
    end for
    Calculate Q_i in Eq.(23).
end for
Q = Q_{T_1}
```

B Supplementary Experiment

B.1 Clustering Degeneracy Study

We conducted comparative experiments to verify whether our method can prevent the occurrence of the clustering degeneracy problem. Clustering degeneracy is a significant challenge for imbalanced datasets (i.e., although the number of categories is provided to the model during training, the predicted number is still smaller than the real amount).

The results are shown in Figure 5. From these results, we can observe that, IOCC converges to the real category number, while other methods suffer from the clustering degeneracy problem.

B.2 The visualization of text representations

To observe the distribution of samples in the feature space, we performed t-SNE visualization on SearchSnippets dataset for baseline models and IOCC. The result is shown in Figure 6. We can see that: (1) In **M3**, all the clusters overlap with each other. (2) **RSTC** shows some improvement over M3, but still contains a significant number of misclustered noise points, indicating poorer clustering performance. (3) **COTC** achieves a better representation distribution than RSTC, but it still has some errors, particularly confusing the clusters represented by red color and black color. (4) Our proposed **IOCC** achieves the best clustering performance. It effectively reduces the noise points within the clusters obtained by clustering. The representation visualization indicates that our proposed method learned discriminative representations and achieved better clustering.

B.3 The Comparison Results Using the Same Encoder

To ensure a fair comparison of algorithm performance, additional experiments were conducted using a unified Encoder. Among the baseline models, SCCL (Zhang et al., 2021), RSTC (Zheng et al., 2023), and COTC (Li et al., 2024) utilize the *distilbert-base-nli-stsb-mean-tokens* (SBERT) Encoder, MIST (Kamthawee et al., 2024) employs the *paraphrase-mpnet-base-v2* (MPNET) Encoder, and STSPL-SSC (Nie et al., 2024) uses the *bge-base-en-v1.5* (BGE-M3) Encoder. Notably, SBERT yields the lowest performance, MPNET surpasses SBERT, and BGE-M3 produces the best results.

In real-world short text clustering applications, the primary objective is to achieve the most accurate clustering results. To this end, IOCC adopts the same BGE-M3 Encoder used by STSPL-SSC (Nie et al., 2024). Different encoders may yield varying results; therefore, to ensure a fair comparison with previous studies, we replaced the encoders for IOCC and baseline models with the BGE-M3

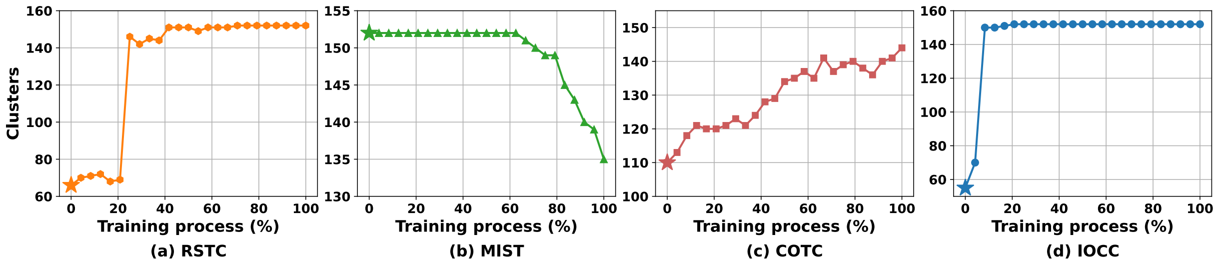

Figure 5: **Clustering Degeneracy Comparison.** The number of predicted clusters during the training process on the GoogleNews-T dataset.

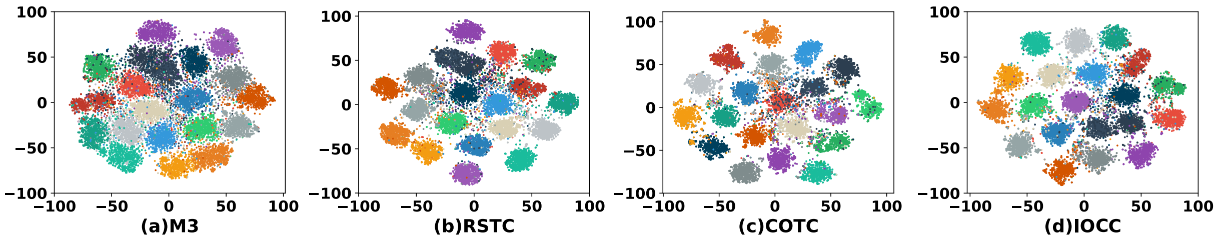

Figure 6: **t-SNE Comparison**. Each color indicates a truth category.

Encoder and SBERT Encoder, respectively.

The results, presented in Table 5 & 6, indicate that under identical Encoder conditions, IOCC continues to outperform the other models. Therefore, the superior performance achieved by IOCC is not closely related to the encoder.

B.4 Research on Incorporating Labeled Data

Like the previous work STSPL-SSC, IOCC is a semi-supervised approach, while the other previous works are unsupervised methods. To ensure a fair comparison, we incorporated the same amount of labeled data used in IOCC into the previous works and applied the cross-entropy loss function to leverage the labeled data.

The results, presented in Table 7, indicate that simply incorporating a small amount of labeled data does not improve model performance. In fact, it has a negative impact. We attribute this to the fact that previous works utilize k-means to generate pseudo-labels at the beginning of the training process. K-means assigns random labels to the generated clusters, which may conflict with the true labels. Furthermore, these results demonstrate that the strong performance of our method is not solely due to the labeled data, but rather to its ability to effectively propagate knowledge from the labeled data to the unlabeled data.

C Computation Budget

We built our model using PyTorch and performed all experiments on an NVIDIA GeForce RTX 3090 Ti GPU. To provide a comprehensive comparison with prior research, we evaluate both the parameter count and training time relative to existing methods, using the StackOverflow dataset as a benchmark. This comparison offers insights into the computational efficiency and scalability of our approach in relation to previous studies.

The results in Table 8 show that, due to the adoption of BGE-M3 as the Encoder, our model has over more 40M parameters compared to RSTC-origin and COTC-origin. However, this increase is negligible relative to the significant improvement in clustering performance. Additionally, in previous work, MIST also uses a new Encoder, making its parameter count comparable to ours, but its clustering performance is still significantly lower than IOCC (as shown in Table 2). Furthermore, IOCC achieves the shortest training time except for RSTC-origin, indicating lower computational resource requirements. When RSTC and COTC are switched to BGE-M3 Encoder, their parameters and training time increase substantially.

D Hyperparameter Analysis

We conducted a series of experiments to validate the effects of ε_1, ε_2, ε_3 and λ with values in $\{0, 1, 5, 10\}$, $\{0.03, 0.06, 0.1, 1, 3.5, 7, 10, 100\}$, $\{10, 15, 20, 25, 30\}$ and $\{1, 5, 10, 15, 20\}$, respectively. The experiments were conducted on the representative datasets **AgNews**, **GoogleNews-T** and **Tweet**. The experimental results are presented in Figure 7.

Method	Agn	Sea	Sta	Bio	GN-TS	GN-T	GN-S	Twe
RSTC	89.39	81.26	86.78	51.67	84.21	80.12	82.82	77.06
STSPL-SSC	89.92	81.04	86.74	47.43	84.41	81.01	82.30	79.59
COTC	88.33	89.78	89.83	51.92	89.56	85.02	87.10	91.53
IOCC	**90.28**	**90.44**	**90.38**	**60.54**	**92.92**	**87.71**	**87.64**	**92.11**
Improvement	**+0.36**	**+0.66**	**+0.55**	**+8.62**	**+3.36**	**+2.69**	**+0.54**	**+0.58**

Table 5: **Results of Using the Same BGE-M3 Encoder.** The experiment results for baseline models using the same BGE-M3 Encoder.

Method	Agn	Sea	Sta	Bio	GN-TS	GN-T	GN-S	Twe
SCCL	83.10	79.90	70.83	42.49	82.51	69.01	73.44	73.10
RSTC	84.24	80.10	83.30	48.40	83.27	72.27	79.32	75.20
COTC	87.56	**90.32**	87.78	53.20	90.50	83.53	86.10	91.33
IOCC	**87.73**	90.24	**89.06**	**58.33**	**91.71**	**85.39**	**86.91**	**91.62**
Improvement	**+0.17**	**-0.08**	**+1.28**	**+5.13**	**+1.21**	**+1.86**	**+0.81**	**+0.29**

Table 6: **Results of Using the Same SBERT Encoder.** The experiment results for baseline models using the same SBERT Encoder.

Method	Agn	Sea	Sta	Bio	GN-TS	GN-T	GN-S	Twe
RSTC	84.76	79.55	81.89	45.31	80.91	70.99	77.89	70.55
MIST	85.51	75.93	82.20	39.85	86.42	73.22	79.45	87.45
STSPL-SSC	89.92	81.04	86.74	47.43	84.41	81.01	82.30	79.59
COTC	87.06	**90.65**	87.17	52.79	88.70	83.03	84.31	90.14
IOCC	**90.28**	90.44	**90.38**	**60.54**	**92.92**	**87.71**	**87.64**	**92.11**
Improvement	**+0.36**	**-0.21**	**+3.21**	**+7.75**	**+4.22**	**+4.68**	**+3.33**	**+1.97**

Table 7: **Results of Incorporating Labels for Baselines.** The comparison between IOCC and previous models with labeled data incorporated.

	RSTC-origin	RSTC-M3	COTC-origin	COTC-M3	MIST-origin	**IOCC**
Training time	00:15:39	00:28:40	00:35:21	01:02:36	00:37:27	00:24:01
Parameters	68.25M	111.37M	77.44M	120.55M	109.5M	111.37M

Table 8: **The Comparison of Parameter Quantity and Training Time.** Where "RSTC-origin", "COTC-origin" and "MIST-origin" refer to the models presented in their respective original papers, while "RSTC-M3" and "COTC-M3" denote the models with the Encoder replaced by BGE-M3.

From Figures 7(a), 7(c), and 7(d), we observe that variations in ε_1, ε_3, and λ have minimal impact on model performance, suggesting that the model is largely insensitive to these parameters. In contrast, Figure 7(b) emphasizes the importance of tuning ε_2 for imbalanced datasets, whereas it has no discernible effect on balanced datasets. Since ε_2 regulates the penalty strength for the imbalance levels of predicted cluster probabilities in Eq.(11), we determine its value based on the degree of imbalance in the dataset.

Although our model has several hyperparameters, only ε_2 influences the performance on imbalanced datasets. This suggests that the model exhibits strong robustness and generalizability. Consequently, when applied to unseen data, the model demonstrates higher adaptability, requiring minimal hyperparameter tuning for effective performance. Experientially, we set $\varepsilon_1 = 1$, $\varepsilon_3 = 25$ and $\lambda = 5$ for all datasets; $\varepsilon_2 = 1000$ and 1.2 for balanced datasets and severely imbalanced datasets, respectively.

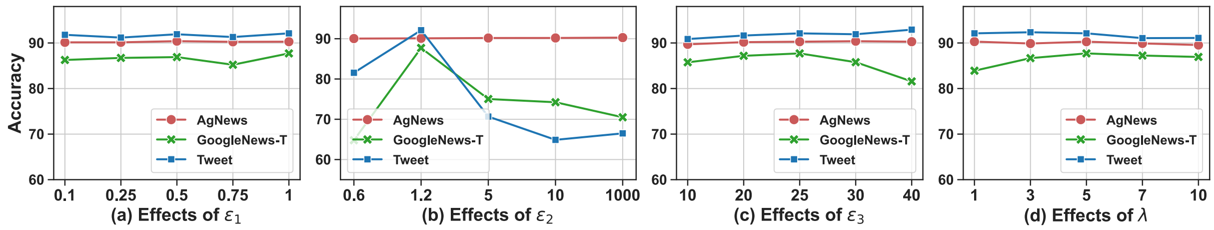

Figure 7: **Hyperparameter Analysis.** The effect of ε_1, ε_2, ε_3, and λ on model accuracy.

E Supplementary Details

E.1 Pseudocode of IOCC

We present the pseudocode of IOCC's training process for an iteration, as shown in Algorithm 2.

Algorithm 2 Pseudocode for an iteration of **IOCC**

Input: Encoder f; Classifier g; Projector h; Mini-batch labeled data $\{X^{l(0)}, Y^l\}$; Mini-batch unlabeled data $X^{u(0)}$; current iteration *iter*.

Output: Updated parameters

```
# generate augmented samples
X^{l(1)}, X^{l(2)} ← textual augmenter(X^{l(0)})
X^{u(1)}, X^{u(2)} ← textual augmenter(X^{u(0)})
# forward texts and obtain P and Z
P^{l(1)}, P^{l(2)}, P^{u(0)}, P^{u(1)}, P^{u(2)} ← f(g(∼))
Z^{l(0)}, Z^{u(0)}, Z^{u(1)}, Z^{u(2)} ← f(h(∼))
# produce pseudo-label via IEOT
Ŷ^u ← IEOT(P^{u(0)})                                   # Eq.(3)
# accumulate and update pseudo-center
η ← 𝟙(max(P^{u(0)}) ≥ τ)
C̄ ← accum. pseudo-center(Z^{l(0)}, Z^{u(0)}, η)
C ← update pseudo-center(C̄)                            # Eq.(4)
# calculate the loss function
L_I ← calculate loss(Z^{u(1)}, Z^{u(2)})                # Eq.(7)
L_X ← calculate loss(P^{l(1)}, P^{l(2)}, Y^l)          # Eq.(9)
L_C ← calculate loss(P^{u(1)}, P^{u(2)}, Ŷ^u)          # Eq.(8)
L ← L_I + L_X + L_C                                     # Eq.(10)
if iter ≥ E_first then
    L_P ← calculate loss(Z^{u(1)}, Z^{u(2)}, Ŷ^u, C)
                                                        # Eq.(5)
    L ← L + λL_P                                        # Eq.(10)
end if
# update parameters
back propagation(L)
```

E.2 Datasets

We conduct experiments on eight benchmark datasets, which cover a wide range of text sources, including news headlines and social media content. These diverse sets enable a thorough evaluation of the model across various domains. Based on the degree of imbalance, **AgNews**, **StackOverflow**, and **Biomedical** are classified as balanced datasets, while **SearchSnippets** is categorized as a slightly imbalanced dataset. In contrast, **GoogleNews-TS**, **GoogleNews-T**, **GoogleNews-S**, and **Tweet** are considered as severely imbalanced datasets. The brief descriptions are provided below:

- **AgNews**: Sourced from AG's news corpus (Zhang et al., 2015), this dataset contains 8,000 news headlines categorized into four different topics (Rakib et al., 2020).
- **SearchSnippets**: Derived from web search activities, it includes 12,340 search result snippets organized into eight distinct categories (Phan et al., 2008).
- **StackOverflow**: Comprising 20,000 question titles across 20 technical fields (Xu et al., 2017), this dataset is sampled from Kaggle competition data, covering technical discussions and programming-related queries.
- **Biomedical**: This dataset consists of 20,000 research paper titles in 20 scientific disciplines (Xu et al., 2017), sourced from BioASQ, showcasing the specialized terminology and format typical of academic research.
- **GoogleNews**: Providing a broad range of news content, it includes 11,109 articles related to 152 events (Yin and Wang, 2016). The dataset is available in three versions: complete articles (GoogleNews-TS), titles only (GoogleNews-T), and snippets only (GoogleNews-S).
- **Tweet**: Containing 2,472 tweets linked to 89 different queries (Yin and Wang, 2016), this dataset was gathered from the Text Retrieval Conference's microblog tracks in 2011 and 2012, reflecting the casual and succinct nature of social media posts.

E.3 Evaluation Metrics

Consistent with previous works (Rakib et al., 2020; Zheng et al., 2023), we employ two standard metrics to use the clustering performance: Accuracy (*ACC*) and Normalized Mutual Information (*NMI*). Accuracy measures the proportion of correct clustered texts, which is defined as:

$$ACC = \frac{\sum_{i=1}^{N} \mathbb{1}_{y_i = \text{map}(\tilde{y}_i)}}{N}, \tag{33}$$

where y_i is the true label and $\tilde{y}_i$ is the predicted label, $\text{map}(\cdot)$ operation refers to aligning the predicted labels with the true labels using the Hungarian algorithm. (Papadimitriou and Steiglitz, 1998).

Normalized Mutual Information quantifies the shared information between the true and predicted label distributions, normalized by their individual uncertainties:

$$NMI(\boldsymbol{Y}, \tilde{\boldsymbol{Y}}) = \frac{I(\boldsymbol{Y}, \tilde{\boldsymbol{Y}})}{\sqrt{H(\boldsymbol{Y})H(\tilde{\boldsymbol{Y}})}} \tag{34}$$

where $\boldsymbol{Y}$ and $\tilde{\boldsymbol{Y}}$ represent the true and predicted label matrices respectively, I denotes mutual information, and H represents entropy.

E.4 Experiment Settings

The batch size of the labeled and unlabeled data are set to $B = 15$ and $\mu B = 200$, respectively. The temperature parameters for instance-wise and prototypical-based contrastive learning are set to $T_P = 1$ and $T_I = 1$. The outer loops of the Majorization-Minimization algorithm T_1 and the iterations of the Lagrange multiplier algorithm T_2 are set to 10. The total number of training iterations E_{total} is 1,500 for all datasets except the Tweet dataset, where $E_{total} = 1,000$. The number of first stage iterations E_{first} is 1,000 for all datasets except the Tweet dataset, in which $E_{first} = 700$. The maximum sentence length of the Encoder f input is 32. The output dimension of the Projector h is set to $D = 128$.